An Interior Point Method Applied to Flow Constraints in a Pressure-Dependent Water Distribution System

Sylvan Elhay, Olivier Piller, Jochen Deuerlein, Angus Simpson

An interior point method applied to flow constraints in a pressure dependent water distribution system

Sylvan Elhay*[1] Olivier Piller[2] Jochen W. Deuerlein[3]
Angus R. Simpson, M.ASCE[4]

October 5, 2021

Abstract

In this paper, the fusion of an active set method (ASM) and an interior point method (IPM) is used, with a Newton method, to solve for the steady-state flows, heads and outflows of a pressure dependent water distribution system. The outflow constraints which arise from the pressure dependency are handled by an ASM and the linkflow constraints are handled by an IPM. The authors believe that this is the first time an ASM and an IPM have been used together in this way to solve a real world optimization problem. Including flow constraints in a network model allows a variety of flow control devices (flow control valves, check valves, pumps) to be modelled efficiently. The new method does not require damping. The separate treatment methods for the two constraint sets means that the linear inequality constraint qualification condition cannot be violated during iteration, unlike the case where all the constraints are handled by an ASM. The method is shown to quickly converge on nine case study networks the largest of which has more than 157,000 links and 150,000 nodes and 6,000 linkflow constraints. When tested on those same nine networks, the popular interior point optimizer package IPOPT is shown to take between about 2–4 times as long as the new method for 5 of the networks, 7 and 9 times as long for 2 networks and about equal time on the 2 remaining networks. For the largest network IPOPT takes 34 minutes while PHYAI takes 4 minutes. The generality of the method means that it has application to network design and management, capacity analysis and self-cleaning networks.

Keywords: Networks, convex programming, interior point methods, active set methods, flow control

[1]Visiting Research Fellow, School of Computer Science, University of Adelaide, South Australia, 5005, sylvan.elhay@adelaide.edu.au. Corresponding author.

[2]Senior Research Scientist, INRAE, ETBX Research Unit, Aqua Department, 50 Ave. de Verdun, Gazinet, F-33612 Cestas, France, olivier.piller@inrae.fr & Adjunct Senior Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide, South Australia, 5005.

[3]Senior Researcher, 3S Consult GmbH, Albtalstrasse 13, D 76137 Karlsruhe, Germany deuerlein@3sconsult.de & Adjunct Senior Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide, South Australia, 5005.

[4]Professor Emeritus, School of Civil, Environmental and Mining Engineering, University of Adelaide, South Australia, 5005. angus.simpson@ adelaide.edu.au

INTRODUCTION

Water Distribution Network Modelling Background

The traditional demand dependent models (DDMs) ((Cross 1936), (Martin & Peters 1963), (Wood & Charles 1972), (Todini & Pilati 1988)) of water distribution systems (WDSs) deliver the nominal nodal demands regardless of the pressures in the system. This can lead to model solutions which are mathematically correct but which are not physically realizable. For example, if the pressure at a node falls below a certain level in a real network, then the nominal demand cannot be realized. As a consequence, attention has turned to pressure dependent models (PDMs) of WDSs and much research effort has been directed at finding powerful and robust solvers to quickly find the steady-state linkflows, outflows and heads of real PDM WDSs.

PDM models require a pressure outflow relationship (POR) which describes the outflow at each demand node as a function of available pressure: nodes at which the pressure is below a preset minimum level, h_m, have zero outflow while those at which the available pressure is above a maximum preset service pressure, h_s, will have the nominal demand delivered. The POR describes the proportion of the nominal demand that is delivered for pressures between h_m and h_s. The interested reader is directed to Sayyed et al. (2015) or Deuerlein et al. (2019) for discussions of some commonly used PORs .

Both Bhave (1981) and Tabesh (1998) proposed solving the PDM problem by using a two-step iterative procedure called Node Flow Iteration. Here the problem is first solved as a DDM and then the demands are corrected according to a chosen PDM relationship. More recently, Jun & Guoping (2013) proposed a solution technique in which the PDM problem is solved by repeatedly solving the corresponding DDM problem by the Global Gradient Algorithm (GGA) of Todini & Pilati (1988) in an extension to the widely-used WDS DDM solver, EPANET (Rossman 2000) and adjusting the demands after each solution. Lippai & Wright (2014) introduced artificial reservoirs, artificial check valves and artificial flow control valves to prevent the outflows from lying outside a prescribed interval. Elsewhere, Piller & van Zyl (2014*a*) used the power equation or the Fixed and Variable Area Discharge (FAVAD) pressure-dependent leakage equation at nodes with leakage to model the dependence of flow on pressure. A review of many of the available approaches for simulating or emulating PORs can be found in Suribabu et al. (2019).

The PDM problem can alternatively be viewed as an optimization problem with linear constraints on the outflows at the nodes and the authors used this approach to solve the unconstrained primal-dual and dual problems by a Newton method with line search to find the heads, linkflows and outflows of a PDM WDS (Elhay et al. 2016). The authors later developed (Deuerlein et al. 2019) a *content-based* active set method (ASM) for the PDM convex optimization problem

$$\min_{\boldsymbol{q},\boldsymbol{c}} f(\boldsymbol{q},\boldsymbol{c}) \mid \boldsymbol{A}^T\boldsymbol{q} + \boldsymbol{c} = \boldsymbol{o},\ g_i(\boldsymbol{c}) \leq 0,\ i = 1, 2, \ldots, m_g, \tag{1}$$

where the *content* function, f, is strictly convex, $\boldsymbol{A}$ is a node-arc incidence matrix, the g_i define m_g linear constraints and $\boldsymbol{q}$ and $\boldsymbol{c}$ are the vectors of decision variables representing the network link flows and nodal outflows, respectively. The resulting ASM is robust and rapid and well-suited for PDM WDSs. Indeed, the Newton-type methods used in the ASM are particularly suitable for WDS models because even though the equations involved are non-linear, they are very structured and sparse and so even models that have hundreds of thousands of degrees of freedom can be quickly and conveniently solved on modern desktop computers. In addition, using the inverse POR , $h(c)$, as the authors did, brings a computational advantage: the Darcy-Weisbach head loss model is quadratic and pseudo-head loss function, $h(c)$, for the Wagner (and any similar POR) is also superlinear. The consequence is that, although in similar circumstances damping or regularization might be necessary, in this case they are not because the convergence ball for the problem is larger and as a result good convergence is more easily achieved.

Addition of Controls

However, when combinations of throttle control valves (TCVs), check valves (CHVs), flow control valves (FCVs), pressure sustaining valves (PSVs), pressure reducing valves (PRVs) or pumps are added to the models, new difficulties arise (Deuerlein et al. 2012). In some cases the methods fail to converge (see Simpson (1999) and Deuerlein et al. (2008)) or even converge to the wrong solution (Gorev et al. 2016). These difficulties are compounded by the fact that in many cases there is no indication that the computed solution is in fact wrong or not physically realizable.

Control devices are increasingly used in WDSs. Limited resources during hot summers, contamination of ground water and the need for more flexible and resilient systems are only a few of the drivers for this development. There is no doubt that the planning and operation of modern systems using smart devices require robust and efficient hydraulic modelling software.

An FCV constrains the flow in a link to be below some chosen value, q_{max}, that is independent of the difference between the heads at the link's initial and final nodes. If the flow is below q_{max}, the valve is opened fully and behaves like a minor loss element. If conditions are such that the unrestricted flow would be higher than q_{max}, the FCV acts to restrict the flow. It is natural then to model the action of such a device by a mathematical constraint. Applying linkflow constraints to the minimization of the content associated with a WDS is then a constrained optimization problem and this formulation of the problem provides some interesting returns: the characterisation of the conditions under which solutions exist and are unique can be studied and a wide range of optimization techniques, including ASMs and interior point methods (IPMs), can be brought to bear on the problem.

Formulation of the Problem Addressed

A number of different approaches to the solution of flow constrained problems have

been reported. Piller & van Zyl (2014*b*) used quadratic external penalty methods applied to the valves' head loss equations. The penalty terms are interpreted as a minor head loss. Later, Alvarruiz et al. (2018) and Rossman et al. (2020) used linear external penalty methods in the loop method to model flow regulating devices. The external penalty terms in these methods generate very large resistance factors and, when these exist alongside normal resistance factors in the models, the resulting matrices can have very large condition numbers. This leads, in many cases, to the need for damping or regularization schemes to be incorporated into the methods in order to avoid divergence.

In Piller et al. (2020) the authors presented a new content-based active set method, ASMFC, for the determination of linkflows, nodal heads and outflows in a PDM WDS which has linkflow constraints. This time the problem addressed was

$$\min_{\boldsymbol{q},\boldsymbol{c}} f(\boldsymbol{q},\boldsymbol{c}) \mid \begin{cases} \boldsymbol{A}^T\boldsymbol{q} + \boldsymbol{c} = \boldsymbol{o}, \\ g_i(\boldsymbol{c}) \leq 0, \ i = 1,2,\ldots,m_g, \\ s_j(\boldsymbol{q}) \leq 0, \ j = 1,2,\ldots,m_s \end{cases} . \tag{2}$$

The necessary and sufficient Karush-Kuhn-Tucker (KKT) conditions for the unique minimum of the content function, f, optimization problem were given and a Newton method based on this formulation led to the ASMFC. The method was demonstrated by applying it to eight case study networks, N_1–N_8, with between 934 and 19,647 links and between 848 and 17,971 nodes. These case study networks each had 60 cotree linkflow constraints which either (i) limit the maximum flow in a link to be between 0 and $\pm 10\%$ of its unconstrained value or (ii) prescribe a link flow equality constraint (LFEC), i.e. the flow in link j is set by requiring $q_{max,j} = q_{min,j}$. The method rapidly found the solutions to all eight case study networks. An advantage of the method is that FCVs and pumps can be modelled and heuristics are not needed to determine the status of control devices in the system: their states are found as part of the solution. Furthermore, there is no risk of isolated demands in this case since in PDM problems the nodal outflow can reduce to zero if the pressure is insufficient.

In some settings flow velocities are required to be above a certain lower limit. Self-cleaning networks (Abraham et al. 2018), for example, limit the flow capacity in all the network's cotree (by closing some valves) to ensure higher velocities in the spanning tree links. Interest in this area led the authors to conduct investigations into the performance of the ASMFC on problems like these. In these new tests the eight case study networks had constraints applied to (almost) every link in each network's cotree (rather than limiting the number of constraints to 60). Confining the constraints to the network cotrees ensured that no parts of the networks were isolated from sources. This led, for the largest network, to $\approx$ 1700 constraints being imposed. It was found that the ASMFC converged rapidly for six of the eight networks but for networks N_4 and N_7, convergence with the ASMFC remained elusive. Now, it is known (see Piller et al. (2020) for details) that, provided certain conditions which are known to hold in this case are met, the solution to this problem always exists and is unique. The problem to finding these solutions, it emerged,

was poor starting values - a common problem in the numerical solution of nonlinear systems. If, on these more challenging problems, the ASMFC is started at a point close to the true solution it reliably converges quite rapidly. But without good starting values stalling and cycling (failure to converge) were observed.

Damping, a technique used by the authors in Elhay et al. (2016) to overcome convergence failures in PDM problems, has been helpful where initial values are far from the solution or where link characteristics are modelled by sublinear functions or even when very large and very small resistance factors in the same models lead to poor matrix conditioning. It has, however, the disadvantage that the co-energy functions (content, co-content or Lagrangian) need to be frequently evaluated and these add an undesirable computational burden. This led the authors to investigate alternative methods and resulted in the new hybrid method of this paper, HYAI.

The Solution

The difference between the problems defined in (1) and (2) turns, of course, on the addition of the linear inequalities $s_j(\boldsymbol{q}) \leq 0$ (the linkflow constraint set) and seeing that those constraints are not always well-handled by the ASM, an alternative way of handling them was found. The constraints $s_j(\boldsymbol{q}) \leq 0$ in (2) could theoretically be implemented by omitting those constraints from the definition of the problem and adding an indicator function term $I_-(s_j(\boldsymbol{q}))$ to the objective function. Here, $I_-(\cdot)$ is defined by

$$I_-(x) \stackrel{\text{def}}{=} \begin{cases} 0 & x \leq 0 \\ \infty & x > 0 \end{cases}. \tag{3}$$

Log barrier IPMs replace the function $I_-(\cdot)$ in (3) by a logarithmic penalty function, $\frac{1}{t}\phi(q)$, which drives the optimization away from the constraint boundaries while keeping them in the interiors of the constraint intervals. The key element in the HYAI method, and the reason it is said to be hybrid, is that it uses a log barrier penalty to enforce the linkflow constraints while continuing to use the ASM approach to handling the POR constraints (via the inverse POR function) on the nodes. The new method is denoted by HYAI to suggest the hybridization of an ASM and an interior point method (IPM). The method borrows from the theory of IPMs for dealing with the linkflow constraints but cannot properly be said to be an IPM in view of its use of the ASM. It will be seen that implementing HYAI requires only a small modification of the ASMFC method.

The penalty terms in HYAI are such that the barrier terms more closely approach $I_-(\cdot)$ as t increases. Thus, larger values of t give solutions which are closer to the exact solution or the ASMFC solution. But finding the minimum of $f(\boldsymbol{x})$ can be difficult with Newton's method when t is large because the Hessian of f varies rapidly near the boundaries of the feasible set (Boyd & Vandenberghe 2009, 564). This difficulty can be overcome by first finding a solution for a small value of t and then solving a sequence of problems in which the solution for each value of t is used to start the Newton method for a larger value of t. This process of continuation can then be used to arrive at a solution for sufficiently large

t.

When a linkflow constraint in an ASM saturates, that effectively causes the link to be removed from the system. Removing a link from the system can, in some instances, cause disconnections in the system's main graph. By comparison, in the hybrid IPM of this paper the full arc-node incidence matrix (ANIM) is always used and some terms are added to the diagonal matrix of head losses. These added terms can assist with convergence. Unlike external penalty methods which sometimes need damping, in this log barrier IPM, damping is avoided by projecting linkflows back to the interior of the constraint interval.

The uniqueness and existence of the problem flows is assured provided the linearly independent constraint qualification (LICQ) is not violated at the solution (Deuerlein et al. 2017) because if the LICQ holds then unique Lagrange multipliers exist and a feasible solution exists. The LICQ requires that the matrix of equality constraints (the mass balance constraints together with the saturated or active constraints) has full rank. For the ASMFC method, the linkflow constraints are part of the LICQ matrix. The violation of the LICQ is equivalent to the isolation of some demand nodes and this can occur if, for example, the linkflow constraints are not confined to the cotree. So, where there are constraints on links in the spanning tree, one should establish (although it is usually impractical to do so) at each iteration that the LICQ is not violated.

The LICQ matrix for the HYAI method is comprised only of (i) the network node-arc incidence matrix, and (ii) the matrix of active outflow constraints, since the linkflow constraints are handled by the IPM, not the ASM. In the Appendix it is shown that the HYAI method LICQ matrix has full rank provided the NAIM has full rank and importantly, that the LICQ matrix and the ANIM of the augmented graph of the network, a key matrix in the method, have the same rank. This fact eliminates a point of failure in the Newton method which is at the basis of the new method. Whether or not a solution exists in the feasible set can be determined by a linear program (see Boyd & Vandenberghe (2009, 579) for details).

Contribution of the New Method

Network designers seek to balance conflicting requirements: (i) the need to satisfy the demands and (ii) the cost of the network construction and maintenance. The search for this balance leads to the need to minimise the head losses, which depend on link diameters and the squares of the velocities (thereby limiting the effects of abrasion at singularities, elbows and shrinkages). A PDM solver with linkflow capacity constraints can be used to help find the optimal solution which balances these requirements. An important problem concerns modelling network capacity: the study of networks for which every link has flow constraints. Capacity problems arise, for example, in transport (communications networks and urban traffic flows) and power systems. A WDS capacity analysis which finds flows that are such that the corresponding velocities are bounded above and below, $v_{min} \leq v \leq v_{max}$ can assist network designers in two important ways. The first is that the lower fluid velocity limit can be set sufficiently high to ensure that the links so constrained

are self-cleaning i.e. have flow velocities high enough to prevent the particle sedimentation that would otherwise occur. For example, in some settings, the fluid velocities in a WDS are required to be limited to no more than 1.5 m/s (sometimes the linkflow constraints are limited to every cotree link and are set to create higher velocities in just the spanning tree links).

The second way that capacity analysis assists the designers stems from the fact that (Boyd & Vandenberghe 2009, 252) the Lagrange multipliers measure the first order sensitivity of the objective function's (in this case the content's) optimal value with respect to perturbations of the constraint. Thus, an examination of the steady-state Lagrange multipliers for the lower velocity bound could indicate how much pumping is necessary in a system and how much extra energy is required to achieve that level of pumping or reduction of head loss.

The authors believe that the HYAI method is the first in which the constraints for some of the problem's objective function variables are handled by using an active set method while others are handled by an interior point method in this way. Its suitability to this approach stems from the fact that the optimization variables naturally separate into two sets: those for the nodes and those for the links.

In this paper the HYAI method is developed and its effectiveness is evaluated and compared to that of ASMFC by its application to the eight case study networks previously used in the evaluation of the ASMFC together with a ninth, much larger network. The new method is shown to solve all the networks considered in this study (some of which previously could not be solved with ASMFC) and its reliability is established by showing the high level of agreement between its solutions and those of ASMFC.

The method is tested on nine networks the largest of which has about 6,000 linkflow constraints: a number which is unrealistic for network control but which is entirely reasonable in the context of design and post-analysis of the resulting outflows and saturation capacities. Moreover, the authors believe that the intensive study of the mathematical background (i) is of value in its own right and (ii) helps provide robust solutions to practical problems. Existing solvers often struggle to find the correct solution even in rather simple cases. It is the objective of the authors to provide a mathematically-based method that is able to find the correct solution for all cases where such a solution exists.

It is shown that, when tested on the nine case study networks, the popular interior point package IPOPT (Wachter & Biegler 2006) takes about 2–4 times as much wall-clock time as PHYAI on 5 networks, 7 and 9 times as long on 2 networks and about the same time on the remaining two networks. In the case of the largest network, IPOPT took about 34 minutes to solve the problem while the new method took about 4 minutes.

The solution of a small example problem is described in detail in the Appendix to help clarify the application of the new method.

The usefulness of the new method, either in its own right or as a method that can be paired with ASMFC, is discussed.

Some summary statistics for four of the nine case study networks which are in the

public domain and summary statistics for the corresponding solution data are provided as supplementary material to the paper.

DEFINITIONS AND NOTATION

Consider a WDS whose network graph has n_p links, or arcs, and $n_j + n_f$ nodes, or vertices: n_j is the number of nodes at which the heads are unknown and $n_f \geq 1$ is the number of source nodes with fixed heads. The links of the network include control valves, pumps and pipes. Denote by $\boldsymbol{q} = (q_1, q_2, \ldots, q_{n_p})^T \in \mathbb{R}^{n_p}$ the vector of unknown flows in the system, $\boldsymbol{h} = (h_1, h_2, \ldots, h_{n_j})^T \in \mathbb{R}^{n_j}$ the unknown heads at the nodes in the system, $\boldsymbol{u} = (u_1, u_2, \ldots, u_{n_j})^T \in \mathbb{R}^{n_j}$ the vector of node elevations and $\boldsymbol{r}(\boldsymbol{q}) = (r_1, r_2, \ldots, r_{n_p})^T$ the vector of link resistance factors. Let $\boldsymbol{A}$ denote the $n_p \times n_j$, full rank, unknown-head ANIM: the ji element of $\boldsymbol{A}$ is (i) -1 if node i is at the end of arc j, (ii) 0 if arc j does not connect to the node i, and (iii) 1 if arc j starts at node i. Let $\boldsymbol{A}_0$ denote the ANIM, with a similar definition, for the fixed-head nodes. Let $\boldsymbol{h}^0$ denote the vector of elevations of the n_f fixed-head nodes. Denote $\boldsymbol{a} = \boldsymbol{A}_0\boldsymbol{h}^0$. Denote by η the exponent used in the head loss formula: $\eta = 2$ for the Darcy-Weisbach model and $\eta = 1.852$ for the Hazen-Williams model. Furthermore, denote by $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^{n_p \times n_p}$ the diagonal matrix whose diagonal elements are defined as $[\boldsymbol{G}(\boldsymbol{q})]_{jj} = r_j|q_j|^{\eta-1}$. Then, $\boldsymbol{\xi}(\boldsymbol{q}) = (\xi_1(q_1), \xi_2(q_2), \ldots, \xi_{n_p}(q_{n_p}))^T = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{q}$ is the vector whose elements model the head losses of the links in the system. In general, (e.g. for the Darcy-Weisbach formula) $\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{q})$ but for the Hazen-Williams formula $\boldsymbol{r}$ is independent of $\boldsymbol{q}$. Denote the vector of the nominal demands at the nodes with unknown-head by $\boldsymbol{d} = (d_1, d_2, \ldots, d_{n_j})^T \in \mathbb{R}^{n_j}$ and denote by n_d the number of nodes with non-zero demands. Denote by $\boldsymbol{c}(\boldsymbol{h}, \boldsymbol{d}) \in \mathbb{R}^{n_j}$ the vector whose elements are the POR function values at the n_j nodes of the system and denote $\boldsymbol{1} = (1, 1, \ldots, 1)^T$. Throughout what follows, the symbol $\boldsymbol{O}$ denotes a zero matrix and $\boldsymbol{o}$ denotes a zero column vector of appropriate dimension for the particular case. The shorthand notation $\boldsymbol{x} + a$, where $\boldsymbol{x}$ is a vector and a is a scalar, will be used to denote the case where every component of $\boldsymbol{x}$ has a added to it. Furthermore, it will be assumed that any matrix inverses which are shown do, in fact, exist. In what follows all the values of t will be powers of 10: $t = 10^e$.

Turning now to PDM problems, assume, for simplicity and without loss of generality, that every node has the same minimum pressure head, h_m, and the same service pressure head, h_s. Individualized minimum and service pressure heads can be implemented by replacing h_m by h_{mi} and h_s by h_{si} throughout but presents no further difficulty. This modification does not change the method and only complicates data management and notation.

Define the pressure fraction, $z(h) \stackrel{\text{def}}{=} (h - (u + h_m))/(h_s - h_m)$. Suppose that $\gamma(\cdot)$ is a bounded, smooth, monotonically increasing function which maps the interval $[0, 1] \to [0, 1]$. The POR at a node is defined by

$$c(h_i) = \begin{cases} 0 & \text{if } z(h_i) \leq 0 \\ d_i\,\gamma(z(h_i)) & \text{if } 0 < z(h_i) < 1 \\ d_i & \text{if } z(h_i) \geq 1 \end{cases}. \tag{4}$$

The inverse function of the POR, the head, $h_i(c)$ expressed as a function of outflow c at node i, will be required for the development of the method. But, the function $h_i(c)$ is not in general everywhere differentiable and so in its place a multivalued, sub-differential mapping is considered:

$$h(c_i) = \begin{cases} \emptyset & \text{if } c_i < 0 \\ (-\infty, h_m + u_i] & \text{if } c_i = 0 \\ (h_s - h_m)\gamma^{-1}\left(\frac{c_i}{d_i}\right) + h_m + u_i & \text{if } 0 < c_i < d_i \\ [h_s + u_i, +\infty) & \text{if } c_i = d_i \\ \emptyset & \text{if } c_i > d_i \end{cases}. \tag{5}$$

Note that, in what follows, careful distinction is made between the scalar variables, h, which represent the heads and the multivalued, sub-differential mapping, $h(c)$ which represents the inverse POR.

METHOD

The problem addressed in this paper is: find the heads, linkflows and nodal outflows in a PDM WDS which has (possibly a large number of) linkflow constraints. The PDM imposes constraints to the nodal outflows and linkflow constraints are introduced in the modelling of flow controls. The constraints are expressed as

$$\begin{pmatrix} \boldsymbol{w}_1(\boldsymbol{c}) \\ \boldsymbol{w}_2(\boldsymbol{c}) \end{pmatrix} \stackrel{\text{def}}{=} \begin{matrix} n_j \\ n_j \end{matrix} \begin{pmatrix} -\boldsymbol{c} \\ \boldsymbol{c} - \boldsymbol{d} \end{pmatrix} \leq \boldsymbol{o}, \quad \begin{pmatrix} \boldsymbol{w}_3(\boldsymbol{q}) \\ \boldsymbol{w}_4(\boldsymbol{q}) \end{pmatrix} \stackrel{\text{def}}{=} \begin{matrix} n_p \\ n_p \end{matrix} \begin{pmatrix} \boldsymbol{q} - \boldsymbol{q}_{max} \\ \boldsymbol{q}_{min} - \boldsymbol{q} \end{pmatrix} \leq \boldsymbol{o}. \tag{6}$$

As a formalism, any links which do not have finite constraints are treated as having $-\infty = q_{min,j} \leq q_j \leq q_{max,j} = \infty$ (in practice only the links which have finite constraints will be treated in these vectors). Define the vector $\phi(\boldsymbol{q})$ whose components are the log barrier terms (Boyd & Vandenberghe 2009, §11.2.1) which constrain the link flows

$$\phi(q_j) = -\ln(-w_j) \tag{7}$$

where j ranges over the components of $\boldsymbol{w}_3$ and $\boldsymbol{w}_4$. Denote the content term

$$\psi(c_i) = \begin{cases} (h_s - h_m)\int_0^{c_i} \gamma^{-1}\left(\frac{x}{d_i}\right) dx, & \text{if } d_i > 0, \\ 0 & \text{if } d_i = 0 \end{cases}$$

denote $\boldsymbol{\psi}(\boldsymbol{c}) = (\psi(c_1), \psi(c_2), \ldots, \psi(c_{n_j}))^T$ and denote
$\boldsymbol{\theta}(\boldsymbol{c}) = \nabla_{\boldsymbol{c}}\boldsymbol{\psi}(\boldsymbol{c}) = (h_s - h_m)\left(\gamma^{-1}(c_1/d_1), \gamma^{-1}(c_2/d_2), \ldots, \gamma^{-1}(c_{n_j}/d_{n_j})\right)^T$. Of central interest here is the restriction of the content function $C(\boldsymbol{q}, \boldsymbol{c})$, where the components of $\boldsymbol{c}$ are defined only on the intervals $0 \leq c_i \leq d_i$.

$$C(\boldsymbol{q}, \boldsymbol{c}) = \sum_{j=1}^{n_p} \int_0^{q_j} \xi_j(s)ds - \boldsymbol{a}^T\boldsymbol{q} + \boldsymbol{c}^T(\boldsymbol{u} + h_m) + \boldsymbol{1}^T\boldsymbol{\psi}(\boldsymbol{c}) \tag{8}$$

More formally, the problem addressed in this paper is:

Problem 0.1 *Find the linkflows, $\boldsymbol{q}$, and outflows, $\boldsymbol{c}$, which minimize the objective function $C(\boldsymbol{q}, \boldsymbol{c})$ subject to the mass balance (equality) constraints*

$$-\boldsymbol{A}^T\boldsymbol{q} - \boldsymbol{c} = \boldsymbol{o} \tag{9}$$

and the (inequality) constraints (6).

The problem, restated using a log barrier technique to restrict the linkflows to the interior of their constraint intervals, is to find

$$\min_{\boldsymbol{q},\boldsymbol{c}} \left\{ C(\boldsymbol{q}, \boldsymbol{c}) + \frac{1}{t}\phi(\boldsymbol{q}^T)\mathbf{1} \right\} \quad \text{subject to } -\boldsymbol{A}^T\boldsymbol{q} - \boldsymbol{c} = \boldsymbol{o} \;\&\; \boldsymbol{o} \le \boldsymbol{c} \le \boldsymbol{d}. \tag{10}$$

(Note that the scalar function ϕ applied to the row vector $\boldsymbol{q}^T$ here produces a row vector result.) The Lagrangian of (10) is

$$\begin{aligned} L(\boldsymbol{q}, \boldsymbol{c}, \boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \textstyle\sum_{j=1}^{n_p} \int_0^{q_j} \xi_j(s)ds - \boldsymbol{a}^T\boldsymbol{q} + \mathbf{1}^T\boldsymbol{\psi}(\boldsymbol{c}) + \boldsymbol{c}^T(\boldsymbol{u} + h_m) - \boldsymbol{h}^T(\boldsymbol{A}^T\boldsymbol{q} + \boldsymbol{c}) \\ + \boldsymbol{\mu}^T(\boldsymbol{c} - \boldsymbol{d}) - \boldsymbol{\lambda}^T\boldsymbol{c} + \tfrac{1}{t}\phi(\boldsymbol{q}^T)\mathbf{1} \\ \text{subject to} \quad \boldsymbol{\lambda} \ge \boldsymbol{o}, \quad \boldsymbol{\mu} \ge \boldsymbol{o}. \end{aligned} \tag{11}$$

Here, $\boldsymbol{h}$, the vector of heads, represents the Lagrange multipliers for the mass balance equality constraint and $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the vectors of non-negative Lagrange multipliers for the inequality constraints on the outflows, $\boldsymbol{c}$.

Denote $\boldsymbol{R} = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_{n_j}\}$ and $\boldsymbol{T} = \operatorname{diag}\{\mu_1, \mu_2, \ldots, \mu_{n_j}\}$. In view of the problem's convexity, the necessary and sufficient conditions for a solution to Problem 0.1 are given by the KKT conditions:

$$\boldsymbol{\xi}(\boldsymbol{q}^*) - \boldsymbol{A}\boldsymbol{h}^* - \boldsymbol{a} + \frac{1}{t}\nabla_{\boldsymbol{q}}\phi(\boldsymbol{q}^*) = \boldsymbol{o} \tag{12}$$

$$\boldsymbol{h}(\boldsymbol{c}^*) - \boldsymbol{h}^* - \boldsymbol{L}^{*T}\boldsymbol{\lambda}^* + \boldsymbol{U}^{*T}\boldsymbol{\mu}^* = \boldsymbol{o} \tag{13}$$

$$-\boldsymbol{A}^T\boldsymbol{q}^* - \boldsymbol{c}^* = \boldsymbol{o} \tag{14}$$

$$-\boldsymbol{L}^*\boldsymbol{R}^*\boldsymbol{c}^* = \boldsymbol{o} \tag{15}$$

$$\boldsymbol{U}^*\boldsymbol{T}^*(\boldsymbol{c}^* - \boldsymbol{d}) = \boldsymbol{o} \tag{16}$$

$$\boldsymbol{\lambda}^* \ge 0, \quad \boldsymbol{\mu}^* \ge 0 \quad . \tag{17}$$

where $\boldsymbol{h}(\boldsymbol{c}^*) = \boldsymbol{\theta}(\boldsymbol{c}^*) + \boldsymbol{u} + h_m$, $\boldsymbol{L}^*$ is a matrix made up of rows of the identity whose indices correspond to those of the nodes at which the lower constraint is saturated (or binding) and $\boldsymbol{U}^*$ is a matrix made up of rows of the identity whose indices correspond to those of the nodes at which the upper constraint is saturated. Relations (15) and (16) represent the complementary slackness conditions, Eq. (12) is the conservation of energy equation and (13) defines $\boldsymbol{h}$ according to (5).

The Newton method for the ASM system without flow control (see Deuerlein et al. (2019) for details) reduces (essentially) to

$$\begin{array}{c} \\ n_p \\ n_j \\ n_b \end{array} \begin{array}{c} \begin{array}{ccc} n_p & n_j & n_b \end{array} \\ \begin{pmatrix} \boldsymbol{F}^{(m)} & -\boldsymbol{A} & \boldsymbol{O} \\ -\boldsymbol{A}^T & \boldsymbol{O} & -\boldsymbol{S}_b^{(m)^T} \\ \widetilde{\boldsymbol{O}} & -\boldsymbol{S}_b^{(m)} & \boldsymbol{M}_b{}^{(m)} \end{pmatrix} \end{array} \begin{pmatrix} \boldsymbol{q}^{(m+1)} - \boldsymbol{q}^{(m)} \\ \boldsymbol{h}^{(m+1)} - \boldsymbol{h}^{(m)} \\ \boldsymbol{c}_b^{(m+1)} - \boldsymbol{c}_b^{(m)} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{\rho}_e^{(m)} \\ \boldsymbol{\rho}_m^{(m)} \\ \boldsymbol{h}(\boldsymbol{c}_b^{(m)}) - \boldsymbol{h}_b^{(m)} \end{pmatrix} \tag{18}$$

where $\boldsymbol{M} = \nabla_{\boldsymbol{c}}\boldsymbol{\theta}(\boldsymbol{c})$ is a diagonal matrix, the diagonal elements of which hold the $\boldsymbol{c}$ derivatives of the inverse POR function $\boldsymbol{h}(\boldsymbol{c})$; the matrix $\boldsymbol{F}$ denotes the diagonal matrix whose diagonal elements are q-derivatives of the corresponding terms in $\boldsymbol{G}(\boldsymbol{q})\boldsymbol{q}$; the energy and mass residuals are defined by $\boldsymbol{\rho}_e^{(m)} = \boldsymbol{\xi}(\boldsymbol{q}^{(m)}) - \boldsymbol{A}\boldsymbol{h}^{(m)} - \boldsymbol{a}$ and $\boldsymbol{\rho}_m^{(m)} = -\boldsymbol{A}^T\boldsymbol{q}^{(m)} - \boldsymbol{c}^{(m)}$; $\boldsymbol{S}_b$ is a selection matrix for the junction nodes associated with constraints that are not saturated and the subscript b indicates that the quantity in question is confined to those components which are associated with an outflow constraint that is not saturated. The index set for these components is, following the notation in Piller et al. (2020), denoted $\mathcal{I}_{c_b}$ and it has n_b elements. Similarly, denote by $\mathcal{I}_q$ the set of indices which represent the links with finite constraints and denote by $n_{\mathcal{I}_q}$ the number of linkflow constraints.

The gradient of the log barrier term in (12) and the Hessian of (7) (which will be incorporated into the equivalent of (18)) are now discussed.

The log barrier terms

From (7), $\nabla_{\boldsymbol{q}}\boldsymbol{\phi} = \sum_j \frac{1}{-w_j}\nabla_{\boldsymbol{q}} w_j$ so $\nabla_q\boldsymbol{\phi} = \boldsymbol{v} = (v_1, v_2, \ldots, v_{n_p})^T$ where

$$v_j = \begin{cases} \frac{1}{(q_j - q_{max,j})} - \frac{1}{(q_{min,j} - q_j)} = \frac{1}{\boldsymbol{w}_{3,j}} - \frac{1}{\boldsymbol{w}_{4,j}} & \text{if } j \in \mathcal{I}_q, \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

In view of the linearity of the constraints, $\nabla_q^2 w_j = 0$ for $\boldsymbol{w}_3$ and $\boldsymbol{w}_4$ so

$$\nabla_q^2\boldsymbol{\phi} = \sum_j \left(\frac{1}{w_j}\nabla_q w_j\right)\left(\frac{1}{w_j}\nabla_q w_j\right)^T. \tag{20}$$

The j-th term in the sum (20) is

$$\frac{1}{(q_j - q_{max,j})^2} + \frac{1}{(q_{min,j} - q_j)^2} = \frac{1}{\boldsymbol{w}_{3,j}^2} + \frac{1}{\boldsymbol{w}_{4,j}^2} \tag{21}$$

so if we denote $\boldsymbol{C}_3 = \operatorname{diag}\{\boldsymbol{q} - \boldsymbol{q}_{max}\}$, and $\boldsymbol{C}_4 = \operatorname{diag}\{\boldsymbol{q}_{min} - \boldsymbol{q}\}$ then

$$\boldsymbol{\Phi} \stackrel{\text{def}}{=} \sum_j \nabla_q^2(q_j - q_{max,j}) + \nabla_q^2(q_{min,j} - q_j) = \begin{pmatrix} \boldsymbol{C}_3^{-2} + \boldsymbol{C}_4^{-2} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \end{pmatrix}. \tag{22}$$

A typical log barrier function $\phi(q)$ and its first two derivatives, $\nabla_{\boldsymbol{q}}\phi(q)$ and $\nabla_{\boldsymbol{q}}^2\phi(q)$, are shown in Fig. 1. Note that $\nabla_q^2\phi(q) > 0$ and, importantly, this establishes that the IPM

preserves strong convexity for this problem and this in turn guarantees that the uniqueness of the solution is preserved.

Setting the gradient of the log barrier Lagrangian (11) to zero

$$\boldsymbol{r}_B = \begin{pmatrix} \boldsymbol{\xi}(\boldsymbol{q}) - \boldsymbol{A}\boldsymbol{h} - \boldsymbol{a} - \frac{1}{t}\boldsymbol{v} \\ -\boldsymbol{A}^T\boldsymbol{q} - \boldsymbol{c} \\ \boldsymbol{h}(\boldsymbol{c}) - \boldsymbol{h} - \boldsymbol{L}^T\boldsymbol{\lambda} + \boldsymbol{U}^T\boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{o} \\ \boldsymbol{o} \\ \boldsymbol{o} \end{pmatrix} \tag{23}$$

gives the system of equations to be solved by the Newton method. It is evident from the form of the first block equation in (23) that the log barrier term can be considered a minor head loss. The Jacobian of this system is

$$\boldsymbol{V} = \begin{pmatrix} \boldsymbol{F} + \frac{1}{t}\boldsymbol{\Phi} & -\boldsymbol{A} & \boldsymbol{O} \\ -\boldsymbol{A}^T & \boldsymbol{O} & -\boldsymbol{I} \\ \boldsymbol{O} & -\boldsymbol{I} & \boldsymbol{M} \end{pmatrix}$$

so the system for the Newton update is now $\boldsymbol{V}^{(m)}\left(\delta\boldsymbol{q}^{(m+1)^T}, \delta\boldsymbol{h}^{(m+1)^T}, \delta\boldsymbol{c}^{(m+1)^T}\right)^T = -\boldsymbol{r}_B^{(m)}$. Denote, dropping superscripts where the context is clear,

$$\begin{aligned} \widehat{\boldsymbol{F}} &= \boldsymbol{F} + \frac{1}{t}\boldsymbol{\Phi} \\ \widehat{\boldsymbol{\rho}}_e &= \boldsymbol{G}(\boldsymbol{q})\boldsymbol{q} - \boldsymbol{A}\boldsymbol{h} - \boldsymbol{a} - \frac{1}{t}\boldsymbol{v} \\ \boldsymbol{\rho}_m &= -\boldsymbol{A}^T\boldsymbol{q} - \boldsymbol{c} \\ \boldsymbol{\rho}_c &= \boldsymbol{h}(\boldsymbol{c}) - \boldsymbol{h} - \boldsymbol{L}^T\boldsymbol{\lambda} + \boldsymbol{U}^T\boldsymbol{\mu} \end{aligned}$$

It is worth noting here that the log barrier terms $\frac{1}{t}\boldsymbol{v}$ and $\frac{1}{t}\boldsymbol{\Phi}$ can be viewed as minor head loss terms and the vector $\frac{1}{t}\boldsymbol{v}$ represents the surrogate Lagrange multipliers for the linkflow constraints at steady state (Boyd & Vandenberghe 2009, 567). As well, it is clear that only a minimal effort is required to modify code to solve a PDM WDS by ASM in order to handle linkflow constraints via this technique. Figure 2 shows a typical function made up of a link head loss, $\xi(q)$, and the corresponding log barrier term $v(q)/t$. The surrogate Lagrange multipliers, ν and κ for head losses h_1 and h_2 are also shown.

The system to be solved is now (cf. (18))

$$\begin{array}{c} \\ n_p \\ n_j \\ n_j \end{array} \begin{array}{c} \begin{array}{ccc} n_p & n_j & n_j \end{array} \\ \begin{pmatrix} \widehat{\boldsymbol{F}} & -\boldsymbol{A} & \boldsymbol{O} \\ -\boldsymbol{A}^T & \boldsymbol{O} & -\boldsymbol{I} \\ \boldsymbol{O} & -\boldsymbol{I} & \boldsymbol{M} \end{pmatrix} \end{array} \begin{pmatrix} \delta\boldsymbol{q} \\ \delta\boldsymbol{h} \\ \delta\boldsymbol{c} \end{pmatrix} = - \begin{pmatrix} \widehat{\boldsymbol{\rho}}_e \\ \boldsymbol{\rho}_m \\ \boldsymbol{\rho}_c \end{pmatrix}. \tag{24}$$

Note that in line with the ASM, the outflow updates are applied to only those nodes in $\mathcal{I}_{c_b}$. Denote by $\boldsymbol{S}_b \in \mathbb{R}^{n_b \times n_j}$ the selection matrix made up of the rows of an n_j identity which correspond to those nodes with positive nominal demand and indices in $\mathcal{I}_{c_b}$. Eq.

(24) can, noting that $n_b \leq n_j$ and that $\boldsymbol{S}_b \boldsymbol{S}_b^T = \boldsymbol{I}_{n_b}$, be rewritten more compactly as

$$\begin{array}{c} \\ n_p \\ n_j \\ n_b \end{array} \begin{array}{c} \begin{array}{ccc} n_p & n_j & n_b \end{array} \\ \begin{pmatrix} \widehat{\boldsymbol{F}} & -\boldsymbol{A} & \boldsymbol{O} \\ -\boldsymbol{A}^T & \boldsymbol{O} & -\boldsymbol{S}_b^T \\ \boldsymbol{O} & -\boldsymbol{S}_b & \boldsymbol{M}_b \end{pmatrix} \end{array} \begin{pmatrix} \delta \boldsymbol{q} \\ \delta \boldsymbol{h} \\ \delta \boldsymbol{c}_b \end{pmatrix} = - \begin{pmatrix} \widehat{\boldsymbol{\rho}}_e \\ \boldsymbol{\rho}_m \\ \boldsymbol{\rho}_{c_b} \end{pmatrix} \tag{25}$$

where $\boldsymbol{M}_b \in \mathbb{R}^{n_b \times n_b}$ is the diagonal matrix with the derivatives of the inverse POR functions for nodes in $\mathcal{I}_{c_b}$ on its diagonal and $\boldsymbol{c}_b \in \mathbb{R}^{n_b}$ is the vector of unsaturated outflows with $d_i > 0$ and $\boldsymbol{\rho}_{c_b}$ is the outflow residual computed only for those outflows which are not subject to active constraints. The system for the hybrid method is now seen to be obtained from the ASM in Deuerlein et al. (2019) by simply adding the log barrier penalty term $\frac{1}{t}\boldsymbol{\Phi}$ to the matrix $\boldsymbol{F}$ on the left and the log barrier term $-\frac{1}{t}\boldsymbol{v}$ to $\boldsymbol{\rho}_e$ on the right. The block equations are

$$\begin{aligned} \widehat{\boldsymbol{F}} \delta \boldsymbol{q} - \boldsymbol{A} \delta \boldsymbol{h} &= -\widehat{\boldsymbol{\rho}}_e & (26) \\ -\boldsymbol{A}^T \delta \boldsymbol{q} - \boldsymbol{S}_b^T \delta \boldsymbol{c}_b &= -\boldsymbol{\rho}_m & \\ -\boldsymbol{S}_b \delta \boldsymbol{h} + \boldsymbol{M}_b \delta \boldsymbol{c}_b &= -\boldsymbol{\rho}_{c_b} & (27) \end{aligned}$$

or

$$\begin{aligned} \boldsymbol{A}^T \delta \boldsymbol{q} - \boldsymbol{A}^T \widehat{\boldsymbol{F}}^{-1} \boldsymbol{A} \delta \boldsymbol{h} &= -\boldsymbol{A}^T \widehat{\boldsymbol{F}}^{-1} \widehat{\boldsymbol{\rho}}_e \\ -\boldsymbol{A}^T \delta \boldsymbol{q} - \boldsymbol{S}_b^T \delta \boldsymbol{c}_b &= -\boldsymbol{\rho}_m \\ -\boldsymbol{S}_b^T \boldsymbol{M}_b^{-1} \boldsymbol{S}_b \delta \boldsymbol{h} + \boldsymbol{S}_b^T \delta \boldsymbol{c}_b &= -\boldsymbol{S}_b^T \boldsymbol{M}_b^{-1} \boldsymbol{\rho}_{c_b} \end{aligned}$$

and adding these three equations gives

$$\left(\boldsymbol{A}^T \widehat{\boldsymbol{F}}^{-1} \boldsymbol{A} + \boldsymbol{S}_b^T \boldsymbol{M}_b^{-1} \boldsymbol{S}_b \right) \delta \boldsymbol{h} = \boldsymbol{A}^T \widehat{\boldsymbol{F}}^{-1} \widehat{\boldsymbol{\rho}}_e + \boldsymbol{\rho}_m + \boldsymbol{S}_b^T \boldsymbol{M}_b^{-1} \boldsymbol{\rho}_{c_b}. \tag{28}$$

Once $\delta \boldsymbol{h}$ is found from the solution of this system, (26) gives

$$\delta \boldsymbol{q} = \widehat{\boldsymbol{F}}^{-1} \left(\boldsymbol{A} \delta \boldsymbol{h} - \widehat{\boldsymbol{\rho}}_e \right) \tag{29}$$

and then (27) gives

$$\delta \boldsymbol{c}_b = \boldsymbol{M}_b^{-1} \left(\boldsymbol{S}_b \delta \boldsymbol{h} - \boldsymbol{\rho}_{c_b} \right). \tag{30}$$

The updated unknowns are found as

$$\boldsymbol{h}^{(m+1)} = \boldsymbol{h}^{(m)} + \delta \boldsymbol{h}^{(m)}, \quad \boldsymbol{q}^{(m+1)} = \boldsymbol{q}^{(m)} + \delta \boldsymbol{q}^{(m)}, \quad \boldsymbol{c}_b^{(m+1)} = \boldsymbol{c}_b^{(m)} + \delta \boldsymbol{c}_b^{(m)} \tag{31}$$

and the updated Lagrange multipliers are found as

$$\boldsymbol{\lambda}^{(m+1)} = -\left(\boldsymbol{h}^{(m+1)} - h_m - \boldsymbol{u}\right), \quad \boldsymbol{\mu}^{(m+1)} = \boldsymbol{h}^{(m+1)} - h_s - \boldsymbol{u} \tag{32}$$

Relations (28)–(32) form the basis of the Newton iterative method. The iterations are run until either (i) the relative difference between the norms of successive iterates is sufficiently small, (ii) too many iterations have been executed, (iii) there has been no reduction in the difference between successive iterates after a prescribed number of steps (stalling), or (iv) The equation residuals are too small to warrant further computation because the correction terms have become too small.

The key matrix in this system is the Schur complement in Eq. (28), $\boldsymbol{A}^T\widehat{\boldsymbol{F}}^{-1}\boldsymbol{A} + \boldsymbol{S}_b^T\boldsymbol{M}_b^{-1}\boldsymbol{S}_b$. The term, $\boldsymbol{S}_b$, represents the ANIM for the system's pseudo links, the links which connect the virtual nodes to a ground node. In the iterative process this term acts like a Levenberg-Marquardt regularization term which helps with convergence. It is revealing to note that the Schur complement can also be written

$$\begin{pmatrix} \boldsymbol{A}^T & \boldsymbol{S}_b^T \end{pmatrix} \begin{pmatrix} \widehat{\boldsymbol{F}}^{-1} & \\ & \boldsymbol{M}_b^{-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{S}_b \end{pmatrix} \tag{33}$$

showing that the right-hand-side matrix is the ANIM for the extended graph.

The linkflows are required to always be in the interior of their constraint intervals so, after the linkflows have been updated, any updates which lie on the boundary of, or outside, their constraint intervals must be projected back into the interiors of their constraint intervals. Setting of the distance, τ_j, from the boundary to the linkflow projection point inside the constraint interval, the projection tolerance, requires some delicacy and is discussed later. The tolerance, τ_j, should be adjusted for the value of t but should never, as a consequence of the precision being used, be so small that the projection is onto the constraint boundary since the log barrier terms in (19) and (21) do not then exist. This is the reason that the HYAI method does not accommodate equality constraints, $q_{min,j} = q_{max,j}$. Where a linkflow is required, moreover, to take a set value this should be handled by treating that flow as known data and not handled as a tightly constrained unknown. In addition, the mass balance should be corrected to take this into account and the corresponding Lagrange multiplier can be found from the energy residuals.

Implementation Details

Starting values

The HYAI method for PDM problems itself requires starting guesses for the linkflows, heads and outflows. The initial HYAI linkflows for unconstrained links or for links with semi-infinite constraints were set to correspond to a velocity of 1/3 m/s. The finitely constrained linkflows were set to the flow rate at the midpoint of the constraint interval. The outflows were set to one half of the nominal demand $d_i/2$. The heads were set to

$h_i(d_i/2)$, the value of the inverse POR for an outflow which is half the nominal demand.

Projection

In IEEE double precision arithmetic, the precision used in all the calculations reported here, $\epsilon_m \approx 10^{-16}$. Taking a projection tolerance which depends only on t, or on t and the length of the constraint interval, for example choosing $\tau = (q_{max} - q_{min})/t$, can lead to undesirable consequences. An example illustrates this point.

Example 1 Suppose $t = 10^{20}$, $q_{max} = 10^{-6}$, $q_{min} = 0$. Then $q_{max} - q_{min} = 10^{-6}$ and $\tau = (q_{max} - q_{min})/t = 10^{-26}$ and the projection point will be computed as $q_{max} - \tau = 10^{-6} - 10^{-26} = 10^{-6}$ in IEEE Double Precision arithmetic. Thus, this linkflow is projected exactly onto the constraint boundary. Using $\tau = 1/t$ does not help because again for any q_{max} larger than about 10^{-4}, $q_{max} - \tau = q_{max}$ in IEEE DP. □

If ϵ_p is a small multiple of the computer arithmetic's machine epsilon, ϵ_m, then projecting to $q_{min,j} + \tau_{min,j}$ on the left boundary and to $q_{max,j} - \tau_{max,j}$ on the right boundary with

$$\tau_{min,j} = \max\left(\frac{1}{t}, \epsilon_p|q_{min,j}|\right), \quad \tau_{max,j} = \max\left(\frac{1}{t}, \epsilon_p|q_{max,j}|\right) \tag{34}$$

ensures that the iterates are always projected to a point close to the boundary (and consistent with the current value of t) but still inside the constraint interval.

Continuation

It was mentioned earlier that IPMs sometimes diverge for the large values of t that are necessary to get sufficient accuracy. Where this happens it may be possible to find a smaller value of t which delivers convergence and then use the solution for that t as the starting value for a larger value of t. Very few iterations are then typically required for the larger values of t. By this process of continuation it is possible to get convergence for large values of t that might diverge with the usual starting values. The stopping test for the HYAI solutions which start the ASMFC method need not use the same small stopping tolerance which is ultimately required in the ASMFC. In the large network examples discussed later, a HYAI stopping tolerance of $\epsilon_s = 10^{-4}$ was found sufficient to guarantee convergence of the ASMFC. This suggests that if the HYAI method is being used on its own and continuation is necessary, then the smaller values of t can be handled with larger stopping tolerances. In the examples discussed below the authors used a sequence of reducing stopping tolerances defined by $\epsilon_s = 10^{10}/t$ to progress from solutions for $t = 10^{12}$ up by factors of 10 to $t = 10^{20}$.

An illustrative example of continuation is given in the next section. It is seen there that the first value of the penalty parameter, $t = 10^{12}$, required six iterations to satisfy the stopping test with $\epsilon_s = 10^{10}/t = 10^{-2}$. For the next value of the penalty parameter $t = 10^{13}$ the stopping tolerance was reset to $\epsilon_s = 10^{-3}$ and convergence took just one iteration. Each of the subsequent values of t up to $t = 10^{20}$, required only one iteration.

Slightly more iterations were typically required for the continuation stages with larger networks. Network N_4, for example, took 35, 5, 1, 2, 5, 1 and 5 iterations for the values of exponent e increasing from 14 to 20, respectively, giving a total of 54 iterations.

There are some practical factors which need to be considered when using continuation: (i) it may need some experimentation to determine the largest value of t for which direct convergence (i.e. convergence without using continuation) is possible (taking t smaller than this value necessarily leads to more iterations than are necessary) and (ii) once that value of t has been obtained it may take some more experimentation to determine the largest factor by which to increase t (again, increasing t by too small a factor un-necessarily increases the number of iterations). Finding a balance between the competing factors here is necessary. The interested reader is referred to Boyd & Vandenberghe (2009, 570 ff.)

Accuracy

A solution, $\widehat{\boldsymbol{x}}(t)$, $t > 0$, to the problem

$$\min_x \left\{ f(\boldsymbol{x}) + \frac{1}{t}\phi(\boldsymbol{x}) \right\} \text{ subject to } g_i(\boldsymbol{x}) \leq 0, i = 1, 2, \ldots, m$$

is called a central point and every central point yields a dual feasible pair which provide a lower bound of the objective function's optimal value v^* (see Boyd & Vandenberghe (2009, 565) for details). From this it follows that $f(\widehat{\boldsymbol{x}}(t)) - v^* \leq m/t$ and that the point $\widehat{\boldsymbol{x}}(t)$ is m/t-suboptimal. Thus, to find a solution with guaranteed accuracy of the objective function ϵ_f requires only that $\widehat{\boldsymbol{x}}(t)$ is a solution for $t = m/\epsilon_f$.

In this study the authors were able to calibrate the accuracy of the HYAI method head, linkflow and outflow solutions by comparing them with those of the ASMFC. Thus, for network N_1, using exponents e = 13–19, the linkflows and heads agree to about 5 digits and the outflows agree relatively to about 10 decimals in infinity norm. At $e = 20$ the linkflows agree to more than 8 decimals, the heads to more than 11 and the outflows to more than 13. These last numbers are about what one would expect with iteration stopping tolerances of $\epsilon_s = 10^{-10}$ and led to the choice of $e = 20$ for all the results reported here on test networks.

Selective Updating and Restarting

Convergence can, in some cases, be assisted by the following simple restarting technique. If the linkflow constraints were being handled by an active set method, three index sets, $\mathcal{I}_{q_l}$, $\mathcal{I}_{q_b}$ and $\mathcal{I}_{q_u}$ would be defined as in Piller et al. (2020). Any linkflows at a constraint boundary for which the Lagrange multipliers are negative would be assigned to the set $\mathcal{I}_{q_b}$, the set of linkflows which are updated during the iteration (linkflows in $\mathcal{I}_{q_l}$ and $\mathcal{I}_{q_u}$ which have non-negative Lagrange multipliers need not be updated since they already satisfy the KKT complementary slackness conditions). Using a strategy which parallels this was found useful to assist and accelerate convergence. A description of that strategy follows.

Recall that here all linkflows are in the interiors of their constraint intervals. Every link for which the flow is sufficiently close to one of its constraint boundaries is identified. Those close to the lower constraint bound are associated with the index set $\mathcal{I}_{q_l}$ and those close to the upper boundary are associated with the index set $\mathcal{I}_{q_u}$. Their corresponding energy imbalances are then computed: $\boldsymbol{\kappa} = \boldsymbol{Gq} - \boldsymbol{Ah} - \boldsymbol{a}$ for the flows in $\mathcal{I}_{q_l}$ and $\boldsymbol{\nu} = -(\boldsymbol{Gq} - \boldsymbol{Ah} - \boldsymbol{a})$ for those in $\mathcal{I}_{q_u}$. The imbalances, $\boldsymbol{\kappa}$ and $\boldsymbol{\nu}$, are approximate, or surrogate, Lagrange multipliers. All links in $\mathcal{I}_{q_l}$ with $\kappa_j < 0$ and links in $\mathcal{I}_{q_u}$ with $\nu_j < 0$ then have their flows reset to the midpoints of their constraint intervals. This strategy can be considered a restarting of the iterative process and was found to assist in attaining rapid convergence. For this purpose the linkflow q_j was considered to be sufficiently close to a constraint boundary if its distance to the boundary was smaller than the tolerance, $\epsilon_s(q_{max,j} - q_{min,j})$, where ϵ_s is a modest multiple of ϵ_p: it was found that ϵ_s needs to be at least one order of magnitude larger than the projection tolerance ϵ_p in order to avoid inconsistent set assignments. The number of iterations required was approximately halved by using restarting in those case study networks where direct convergence was achieved (N_1, N_2, N_3, N_5 and N_6) but where continuation was necessary (N_4, N_7, N_8 and N_9) the number of iterations was about the same. This restarting strategy is therefore recommended.

THE HYAI ALGORITHM

In this section the basic HYAI algorithm is described.

HYAI Algorithm Summary

Compute (i) $\tau_{min,j}, \tau_{max,j}$, for all j and (ii) the initial values $\boldsymbol{q}^{(0)}$, $\boldsymbol{h}^{(0)}$ and $\boldsymbol{c}^{(0)}$.

The iteration loop

For $m = 0, 1, 2, \ldots$ repeat steps (a) to (j) until the stopping test is satisfied

(a) Let n_b denote the number of elements in $\mathcal{I}_{c_b}$. Compute the matrix $\boldsymbol{M}_b^{(m)} \in \mathbb{R}^{n_b \times n_b}$.

(b) $\boldsymbol{\rho}_m^{(m)} \leftarrow -\boldsymbol{A}^T \boldsymbol{q}^{(m)} - \boldsymbol{c}^{(m)}$

(c) Solve (28) for $\delta \boldsymbol{h}^{(m+1)}$ and update the heads $\boldsymbol{h}^{(m+1)}$ using (31).

(d) Use (29) to get $\delta \boldsymbol{q}^{(m+1)}$ and, excluding any constrained links with (i) linkflows which are sufficiently close to a boundary and (ii) have positive Lagrange multipliers, update the pipe flows $\boldsymbol{q}^{(m+1)}$ using (31) .

(e) Use (30) to get $\delta \boldsymbol{c}^{(m+1)}$ and use (31) to update the nodal outflows $c_i^{(m+1)}$ for which $i \in \mathcal{I}_{c_b}$.

(f) $\boldsymbol{\lambda}^{(m+1)} \leftarrow -\left(\boldsymbol{h}^{(m+1)} - h_m - \boldsymbol{u}\right)$

(g) $\boldsymbol{\mu}^{(m+1)} \leftarrow \boldsymbol{h}^{(m+1)} - h_s - \boldsymbol{u}$

(h) Update the index sets $\mathcal{I}_{c_b}, \mathcal{I}_{c_l}, \mathcal{I}_{c_u}$.

(i) $\boldsymbol{c}^{(m+1)} \leftarrow \max(\min(\boldsymbol{c}^{(m+1)}, \boldsymbol{d}), \boldsymbol{o})$ (project any exterior outflows onto the constraint boundaries)

(j) $\boldsymbol{q}^{(m+1)} \leftarrow \max(\min(\boldsymbol{q}^{(m+1)}, \boldsymbol{q}_{max} - \boldsymbol{\tau}), \boldsymbol{q}_{min} + \boldsymbol{\tau})$ (project any linkflows not in the interior of the constraint interval back into the interior)

It is evident that the HYAI method can be easily implemented as a simplification of the ASMFC. The log barrier terms in (21) are added to the first block component of the right-hand-side of the system and the terms shown in (19) are added to the diagonal $\boldsymbol{F}$ matrix. No damping is required and there is no need to be concerned about violating the LICQ.

Algorithmic Variations

The new method can be used in at least three different ways. If direct convergence is achieved for sufficiently large t nothing more is necessary. If, however, direct convergence is not possible for large enough t, then a quickly obtained low precision solution, that can be used in one of two ways, is obtained for the largest value of t that gives direct convergence. That solution can now be used to start the HYAI method for larger t and repeating this process of continuation leads to a solution for the value of t required. Few iterations are required after the first stage of this continuation process. Alternatively, the quickly obtained low precision solution can be used to start the ASMFC method. Again, few iterations are then required by the ASMFC. This second technique, which will be designated PHYAI to suggest a pairing of the methods, turns out to be very effective, particularly on networks that challenge ASMFC. These variations are illustrated and compared in the next section.

EXAMPLES AND DISCUSSION

The new HYAI method and the PHYAI methods were applied to the following problems:
(i) a small illustrative network with $n_p = 10$ links and $n_j = 8$ nodes,
(ii) eight case study networks N_1–N_8 [four of which are available online in the ASCE Library (www.ascelibrary.org)], and which have between 934 and 19,647 links and between 848 and 17,971 nodes and
(iii) a larger network with $n_p = 157,044$, $n_j = 150,630$, and four sources, $n_f = 4$ (Sitzenfrei et al. 2020).
(iv) the small network in Fig. 3. This example is explained in detail in the Appendix to help clarify the process.

The networks in (i) and (ii) were used by the authors in Piller et al. (2020) and their EPANET `.inp` files can be found in the Supplementary Materials to that paper.

Denote the measures of the errors between successive q iterates by

$$\delta_q^{(m+1)} = \max_j \frac{|q_j^{(m+1)} - q_j^{(m)}|}{1 + |q_j^{(m+1)}|}, \tag{35}$$

with corresponding notation for the errors of the c and h iterates. In all cases
(i) the nominal demands were magnified by a factor of 5 to illustrate the behaviour of the methods on PDM problems
(ii) the 1-side regularized Wagner POR was used (see (Deuerlein et al. 2019) for details)
(iii) the mid-interval starting scheme described above was used and
(iv) the iterations were run until the relative differences between successive iterates

$$\delta_q^{(m+1)}, \quad \delta_c^{(m+1)}, \quad \delta_h^{(m+1)} \tag{36}$$

were smaller than the prescribed stopping tolerance $\epsilon_s = 10^{-10}$. This stopping tolerance (even though it is smaller than would normally be used in practice) was chosen to ensure that the quadratic convergence normally associated with Newton's method was evident. The iterations were stopped when (i) the stopping test was satisfied, (ii) the residuals were too small to make any further computation practicable because the iterative correction terms vanished, or (iii) the iteration stalled but left residuals that are sufficiently small. The iteration counts reported below are certainly conservative: practical stopping tolerances, which would be considerably larger than 10^{-10}, would require many fewer iterations than are tabled below. A single iteration of the ASMFC method takes roughly the same wall-clock time as one iteration of the HYAI method.

Except where otherwise stated, all the calculations reported in this paper were performed using the authors' own codes written for Matlab (2020a) which uses IEEE standard double precision floating-point arithmetic. The tests for the case study networks N_1–N_8 were run on a PC with an i7-4700MQ CPU and those for N_9 were run on a PC with an i9-10880H CPU.

Small Illustrative Network Example

Figure 4 shows the small illustrative network and its steady-state solution when all links are essentially unconstrained (none of the constraints are saturated) while Fig. 5 shows the same network when three constraints, all of which are active at steady-state, are applied. The constraints on links 3 and 9 are acting as FCVs while the flow in link 10 is at the lower link flow constraint boundary and, since the corresponding multiplier in this case can be interpreted as a negative head loss, this constraint can be seen to be acting as a pump. The continuation technique is illustrated in Table 1. The initial value of t, which corresponds to an exponent $e = 12$, takes six iterations to satisfy the stopping test and for each subsequent value of t, each increasing by one order of magnitude, takes only one iteration. For the last entry in the table, where $t = 10^{20}$, HYAI and ASMFC solutions agree relatively to about 14 decimal digits but for smaller values of t they match

to fewer digits.

HYAI and PHYAI Methods on Larger Networks

The results of applying both the HYAI method and the PHYAI method (the paired HYAI and ASMFC methods) to the nine case study networks are now discussed. In the case of the PHYAI method there are two cases to consider: (i) direct convergence for the required t or (ii) a continuation solution for the required t. In the case of direct convergence, the HYAI iterations were stopped if either (i) the (more relaxed) stopping test with $\epsilon_s = 10^{-4}$ was satisfied or (ii) more than 12 HYAI iterations had been executed. The solution thus produced was then used to start the ASMFC with $\epsilon_s = 10^{-10}$. Where continuation was required, the graded tolerances scheme which relates the stopping tolerance to the value of t was used to terminate the HYAI stage of the PHYAI. As reported earlier, the HYAI solution for the larger values of t required few iterations.

Columns two to four of Table 2 show the number of links, nodes and sources. Every cotree link for which the unconstrained flow was positive was constrained to $[0, +10\%]$ of that flow while if the flow negative it was constrained to $[-10\%, 0]$ of its unconstrained value. Any constraints for which the width of the constraint interval was smaller than 0.001 L/s was culled from the collection. The number of constraints that remained is shown in column five. Column 6 shows the total nodal delivery as a percentage of the total nominal demand.

Column seven gives the number of HYAI iterations required to satisfy the stopping test (36) with $\epsilon_s = 10^{-10}$ and columns 8–10 show the three quantities $\sigma^h_{q,h,c} \stackrel{\text{def}}{=} -\log_{10} \delta_{q,h,c}$ for the HYAI method. They represent the number of decimal digits of agreement between the last two HYAI iterates as measured by the relative error in (36).

Five of the networks $N_{1,2,3,5,6}$, converged directly with between 12 and 23 iterations for $e = 20$. Networks $N_{4,7,8,9}$, on the other hand, all required continuation starting from $e = 12$ (to achieve an $e = 20$ solution) because, if started with $e = 20$, they had not satisfied the stopping test after more than 70 iterations.

Both HYAI and PHYAI methods took about the same number of iterations for networks $N_{1,2,3,5,6}$ but for networks $N_{4,7,8,9}$ the HYAI method took about twice as many iterations as PHYAI because of continuation. Column 11 shows the number, k^p_h, of iterations required by the HYAI stage of the PHYAI scheme and column 12 shows the number, k^p_a, of iterations required for the ASMFC stage of the PHYAI scheme. Most networks required 3–7 ASMFC iterations, one took 10 and the largest network, N_9, took 18. Column 13 shows the sum, $k^p_h + k^p_a$ and the following three columns show the numbers $\sigma^a_{q,h,c}$ which represent the number of decimal digits of agreement between the last two ASMFC PHYAI iterates as measured by the relative error in (36).

Column 17 shows the quantity α_q which is defined by

$$\alpha_q \stackrel{\text{def}}{=} -\log_{10} \left\{ \max_j \frac{|q_{h,j} - q_{a,j}|}{1 + |q_{a,j}|} \right\}$$

and columns 18 and 19 show the corresponding quantities, α_h for h and α_c for c. These numbers indicate, in decimal digits, the degree of agreement between the solutions of the HYAI method and those of the PHYAI and they are entirely consistent with a stopping tolerance of 10^{-10}.

Thus, the PHYAI is the recommended method where speed is an issue in view of the smaller number of iterations it required. But, one could start by using the HYAI method and, in the event that continuation is required, switch to ASMFC to achieve convergence.

IPOPT AND EPANET ON THESE PROBLEMS

Exactly the same problems used to demonstrate the PHYAI and HYAI methods on networks N_1–N_9 were solved using the Matlab Toolbox written by P. Carbonetto and updated by Bertolazzi (2021) for the off-the-shelf package, IPOPT (Wachter & Biegler 2006). The objective function (8) subject to the equality constraints (9) and the variable bounds (6) was optimized using IPOPT. The comparison with the authors' Matlab codes was performed to provide some indication of the efficiency of the new method.

The 'mumps' linear solver and the 'adaptive' 'mu_strategy' options were used (see `https://coin-or.github.io/Ipopt/OPTIONS.html` for details). In all cases the IPOPT termination status for the stopping tolerance 10^{-10} (the same as was used on the PHYAI and HYAI codes) was shown as "Optimal solution found". The results of these tests are shown in Table 3. Column 2 of Table 3 shows, k_I, the number of IPOPT iterations required to satisfy the termination test and column 3 shows the corresponding PHYAI number, k_P. Column 3 shows the optimal value of the objective function, $C(\boldsymbol{q}, \boldsymbol{c})$, column 5 shows the final delivery fraction, ζ as a percentage. Column 6 shows, s_I, the number of wall-clock seconds for IPOPT to satisfy the stopping test and column 7, shows the number of PHYAI wall-clock seconds to satisfy the stopping test. Column 8 shows the ratio of these times.

The PHYAI codes took fewer iterations than the IPOPT codes in all cases. But an iteration count comparison in a case such as this does not necessarily tell the whole story. Here, wall-clock times are more revealing. The IPOPT method takes about the same wall-clock time as the PHYAI method in two cases, about 2–4 times as long in five others and 7 and 9 times as long in the remaining two cases. For the largest network the savings in real time are significant: about 4 minutes for PHYAI as opposed to about 34 minutes for the IPOPT codes.

The nine test networks were also run with the latest release of EPANET (version 2.2 from July, 23rd, 2020) which includes the simulation of pressure dependent demands. Flow control is, to the knowledge of the authors, treated by valve-state heuristics as in previous versions. In general, the results are promising for the smaller test networks (N_1–N_6). For networks N_7, N_8 and N_9 EPANET fails to converge after 500 iterations. This may be because of the heuristics used for tackling flow control devices. The status report file shows a huge number of state switches over the iterations. Another problem is that EPANET cannot handle positive lower bounds on flows which are required in the design and control of pumps. It is restricted to FCV upper bounds and check valves (one-directional flows,

$q > 0$).

CONCLUSIONS

The authors' ASMFC, which solves for the heads, linkflows and outflows of a PDM WDS, uses an active set method to enforce both the PDM constraints on the outflows and the modelling constraints on the linkflows. The method has been shown to be rapid and robust when applied to eight case study networks, N_1–N_8, with up to 60 linkflow constraints, some of which are equality constraints. Deeper investigations revealed that when every one of the (up to ≈1700) links in the networks' co-trees were constrained to ±10% of their unconstrained values, six of the networks converged rapidly and without difficulty with the ASMFC but two of the networks, N_4 and N_7, failed to converge.

A new hybrid method, HYAI, which combines active set principles and log barrier IPM principles has been presented which overcomes these convergence difficulties: the constraints on outflows inherent in PDM WDSs are handled by an ASM and the linkflow constraints required to model flow control devices are handled by a log barrier IPM. Implementation details concerning projection, continuation, restarting and selective updating have been outlined which improve the performance of the method. The following points are noted.

(a) Unlike external penalty methods HYAI does not require the computational burden of damping but it may require continuation which typically increases the number of iterations.

(b) Whereas with external penalty methods it can be very hard to cause iterates which move far away from the constraint interval back to approach the constraint interval, the HYAI linkflow iterates are always within the constraint intervals.

(c) Implementing HYAI requires only a very simple modification of the ASMFC method.

The new HYAI method and the continuation strategy were demonstrated on a small illustrative network and then used to find the steady-state heads, linkflows and outflows of all eight previously used case study networks and one other, much larger network. The HYAI method achieved direct convergence with five of the nine networks but the other four required continuation. Thus, the HYAI method succeeded in finding steady-state solutions where the ASMFC had not. But on those networks which challenged the ASMFC and some others, the HYAI method on its own took about twice as many iterations as the PHYAI. Pairing the HYAI and ASMFC methods, the PHYAI method, which uses low precision HYAI solutions to start the ASMFC was found to solve the challenging systems in fewer iterations overall. Three different approaches are available: (i) attempt a solution with HYAI seeking direct convergence. If that fails (ii) use a low precision HYAI solution to start the ASMFC or (iii) use the low precision solution to initiate a continuation process that leads to a HYAI solution for the required t. An alternative is to begin by finding a low precision HYAI solution and using that to start ASMFC. This strategy will reduce the

average number of iterations required over a range of networks. The new method was also illustrated in detail on a small example network to help clarify the method for readers.

The same problems that were used to illustrate the new method were solved with the freely available interior point optimization package IPOPT. The IPOPT method takes about the same wall-clock time as the PHYAI method in 2 cases, about 2–4 times as long in 5 others and 7 and 9 times as long in the remaining 2 cases. For the largest network the savings in real time are significant: about 4 minutes for PHYAI as opposed to about 34 minutes for the IPOPT codes.

The authors would, in future work, like to extend this technique to handle pressure control problems and to more deeply investgate the use of capacity analysis in WDS design and in optimal network control.

SUPPLEMENTARY MATERIAL

Network parameter summary statistics and summary solution statistics for those networks which are in the public domain (N_1, N_3, N_4 and N_7) are avaiable from the ASCE as material supplementary to this paper.

DATA AVAILABILITY STATEMENT

Some or all data, models, or code generated or used during the study are proprietary or confidential in nature and may only be provided with restrictions (e.g. anonymized data). The four networks N_2, N_5, N_6 and N_8 are not available either because they are proprietary or because of security concerns.

Some or all data, models, or code used during the study were provided by a third party. (N_9). Direct request for these materials may be made to the provider as indicated in the Acknowledgments.

EPANET `.inp` files for the network shown in Fig. 4 and networks N_1, N_3, N_4 and N_7 are available from the ASCE library (www.ascelibrary.org) as material that is supplementary to Deuerlein et al. (2019).

In addition, EPANET `.inp` files for the network shown in Fig. 3 and networks N_1, N_3, N_4 and N_7 with the flow constraints used in the tests reported here (flow control valves and check valves) are available from the ASCE library as material that is supplementary to this paper.

ACKNOWLEDGMENTS

The authors thank Univ.-Prof. Dipl.-Ing. Dr. Robert Sitzenfrei who provided the authors with the virtRom network, N_9. This network is available from `https://www.uibk.ac.at/umwelttechnik/softwareanddatasets/`. One author gratefully thanks Dr J. Kautsky for useful discussions. One author of this paper was supported in part by the German Ministry for Education and Research (BMBF Project W-Net 4.0 02WIK1477C).

References

Abraham, E., Blokker, E. & Stoianov, I. (2018), 'Decreasing the discoloration risk of drinking water distribution systems through optimized topological changes and optimal flow velocity control', *J. Water Resour. Plann. Manage.* **144**(2), 04017093.

Alvarruiz, F., Alzamora, F. & Vidal, A. M. (2018), 'Efficient modeling of active control valves in water distribution systems using the loop method', *J. Hydraul. Eng.* **144**(10), 04018064. DOI: 10.1061/(ASCE)WR.1943-5452.0000982.

Bertolazzi, E. (2021), 'ebertolazzi/mexipopt', GitHub. github.com/ebertolazzi/mexIPOPT/releases/tag/1.1.2.

Bhave, P. R. (1981), 'Node flow analysis distribution systems', *Transportation Engineering Journal* **107**(4), 457–467.

Boyd, S. P. & Vandenberghe, L. (2009), *Convex Optimization*, 7th edn, Cambridge University Press, UK.

Cross, H. (1936), 'Analysis of flow in networks of conduits or conductors', *Bulletin 266 University of Illinois Engineering Experimental Station* .

Deuerlein, J., Piller, O., Elhay, S. & Simpson, A. R. (2019), 'A content-based active set method for the pressure dependent model of water distribution systems', *J. Water Resour. Plann. Manage.* **145**(1), 04018082. DOI: 10.1061/(ASCE)WR.1943-5452.0001003.

Deuerlein, J., Simpson, A. & Montalvo, I. (2012), Preprocessing of water distribution systems to assess connectivity and solvability in the presence of flow control devices., *in* 'World Environmental and Water Resources Congress 2012', pp. 3237–3247. DOI: 10.1061/9780784412312.325.

Deuerlein, J., Simpson, A. R. & Gross, E. (2008), The never-ending story of modeling control devices in hydraulic systems analysis, *in* J. E. V. Zyl, A. A. Ilemobade & H. Jacobs, eds, 'Proceedings of the 10th Annual Water Distribution Systems Analysis Conference WDSA2008', Kruger National Park, South Africa.

Deuerlein, J. W., Piller, O., Parisini, F., Simpson, A. R. & Elhay, S. (2017), On the solvability of the pressure driven hydraulic steady-state equations considering feedback-control devices, *in* R. Collins, ed., 'Computing and Control for the Water Industry 2017', Research Studies Press ltd. DOI: 10.15131/shef.data.c.3867985.v1.

Elhay, S., Piller, O., Deuerlein, J. W. & Simpson, A. R. (2016), 'A robust, rapidly convergent method that solves the water distribution equations for pressure dependent models', *J. Water Resour. Plann. Manage.* **142**(2). DOI: 10.1061/(ASCE)WR.1943-5452.0000578.

Gorev, N. B., Kodzhespirov, I. F. & Sivakumar, P. (2016), 'Non-unique steady states in water distribution networks with flow control valves', *J. Hydraul. Eng.* **142**(9), 04016029.

Jun, L. & Guoping, Y. (2013), 'Iterative methodology of pressure-dependent demand based on EPANET for pressure-deficient water distribution analysis', *J. Water Resour. Plann. Manage.* **139**(1), 34–44. DOI: 10.1061/(ASCE)WR.1943-5452.0000227.

Lippai, I. & Wright, L. (2014), 'Demand constructs for risk analysis', *Procedia Engineering* **89**, 640–647. 16th Water Distribution System Analysis Conference, WDSA2014 Urban Water Hydroinformatics and Strategic Planning.

Martin, D. & Peters, G. (1963), 'The application of Newton's method to network analysis by digital computer', *Journal of Institution of Water Engineers and Scientists* pp. 115–129.

Matlab (2020a), *version 9.8.0.1359463 (R2020a) Update 1*, The MathWorks Inc., Natick, Massachusetts.

Piller, O., Elhay, S., Deuerlein, J. & Simpson, A. R. (2020), 'A content-based active set method for the pressure dependent model of water distribution systems with flow controls', *J. Water Resour. Plann. Manage.* **146**(4), 04020009.

Piller, O. & van Zyl, J. E. (2014*a*), 'Incorporating the FAVAD leakage equation into water distribution system analysis', *Procedia Engineering* **89**, 613–617. 16th Water Distribution System Analysis Conference, WDSA2014 Urban Water Hydroinformatics and Strategic Planning.

Piller, O. & van Zyl, J. E. (2014*b*), 'Modeling control valves in water distribution systems using a continuous state formulation', *J. Hydraul. Eng.* **140**(11), 04014052.

Rossman, L. (2000), *EPANET 2 Users Manual*, Water Supply and Water Resources Division, National Risk Management Research Laboratory, Cincinnati, OH45268.

Rossman, L., Woo, H., Tryby, M., Shang, F., Janke, R. & Haxton, T. (2020).

Sayyed, M. A. H. A., Gupta, R. & Tanyimboh, T. T. (2015), 'Noniterative application of epanet for pressure dependent modelling of water distribution systems', *Water Resour. Manage.* **29**, 32273242.

Simpson, A. R. (1999), Modeling of pressure regulating devices a major problem yet to be satisfactorily solved in hydraulic simulation, ASCCE. Water Distribution Systems Conference.

Sitzenfrei, R., Wang, Q., Kapelan, Z. & Savi, D. (2020), 'Using complex network analysis for optimization of water distribution networks', *Water Resources Research* **56**(8). doi.org/10.1029/2020WR027929.

Suribabu, C. R., Renganathan, N. T., Perumal, S. & Paez, D. (2019), 'Analysis of water distribution network under pressure-deficient conditions through emitter setting', *Drink. Water Eng. Sci.* **12**, 1–13.

Tabesh, M. (1998), Implications of the Pressure Dependency of Outflows on Data Management, Mathematical Modelling and Reliability Assesment of Water Distribution Systems, PhD, Department of Civil Engineering, University of Liverpool, UK.

Todini, E. & Pilati, S. (1988), *A gradient algorithm for the analysis of pipe networks.*, John Wiley and Sons, London, pp. 1–20.

Wachter, A. & Biegler, L. T. (2006), 'On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming', *Mathematical Programming* **106**(1), 25–57.

Wagner, J., Shamir, U. & Marks, D. (1988), 'Water distribution reliability: simulation methods', *J. Water Resour. Plann. Manage.* **114**(3), 276–294. DOI: 10.1061/(ASCE)0733-9496(1988)114:3(276).

Wood, D. & Charles, C. (1972), 'Hydraulic network analysis using linear theory', *Journal of the Hydraulics Division* **98**(7), 1157–1170.

APPENDIX

UNIQUNESS OF THE SOLUTION AND THE LICQ

The matrix $\widehat{\boldsymbol{C}} = (\,\boldsymbol{A}^T \quad \boldsymbol{S}_b^T\,)$ of (33) has full rank if $\boldsymbol{A}$ has full rank. This fact is established if

$$\begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{S}_b \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} \boldsymbol{o} \\ \boldsymbol{o} \end{pmatrix} \Rightarrow \boldsymbol{x} = \boldsymbol{o}$$

and this is immediate since the first block equation is $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{o}$ and the full rank of $\boldsymbol{A}$ ensures that this is so. In fact the matric $\widehat{\boldsymbol{C}}$ and the LICQ matrix $\boldsymbol{C}$ have the same rank, as can be seen by the following argument. Recall that the HYAI LICQ matrix is

$$\boldsymbol{C} = \begin{array}{c} \\ n_j \\ n_{c_l} \\ n_{c_u} \end{array} \begin{array}{c} \begin{array}{cc} n_p & n_j \end{array} \\ \begin{pmatrix} \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{O} & -\boldsymbol{S}_{c_l} \\ \boldsymbol{O} & \boldsymbol{S}_{c_u} \end{pmatrix} \end{array}$$

Denote $\boldsymbol{S}_a^T = (\,-\boldsymbol{S}_{c_l}^T \quad \boldsymbol{S}_{c_u}^T\,)$. Then, we can rewrite

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{O} & \boldsymbol{S}_a \end{pmatrix}$$

$\boldsymbol{C}$ has full rank if

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{I} & \boldsymbol{S}_a^T \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{o} \\ \boldsymbol{o} \end{pmatrix} \Rightarrow \boldsymbol{x}_1 = \boldsymbol{o},\ \boldsymbol{x}_2 = \boldsymbol{o}.$$

The block equations are $\boldsymbol{A}\boldsymbol{x}_1 = \boldsymbol{o}$ and $\boldsymbol{x}_1 + \boldsymbol{S}_a^T\boldsymbol{x}_2 = \boldsymbol{o}$. Assume these equations hold. Multiplying the second block equation on the left by $\boldsymbol{A}$ gives $\boldsymbol{A}\boldsymbol{S}_a^T\boldsymbol{x}_2 = \boldsymbol{o}$ and $\boldsymbol{x}_2 = \boldsymbol{o}$ iff $\boldsymbol{A}\boldsymbol{S}_a^T$ has full rank. Now, $\boldsymbol{S}_a\widehat{\boldsymbol{C}} = (\,\boldsymbol{S}_a\boldsymbol{A}^T \quad \boldsymbol{O}\,)$ by the orthogonality of $\boldsymbol{S}_a$,and $\boldsymbol{S}_b$. So the ranks of $\boldsymbol{C}$ and $\widehat{\boldsymbol{C}}$ are both determined by the rank of $\boldsymbol{S}_a\boldsymbol{A}^T$: the rows of $\boldsymbol{A}^T$ that are selected by $\boldsymbol{S}_a$ must form a linearly independent set. This will always be true if $\boldsymbol{A}$ has full rank but it must hold if $\boldsymbol{A}$ has less than full rank: the set of active POR constraints must be such that the corresponding rows of $\boldsymbol{A}^T$ form a linearly independent set.

Thus, the LICQ is never violated in the ASM solution of a PDM problem if the NAIM has full rank and there are no linkflow constraints. If LICQ is not violated then unique Lagrange multipliers exist and a feasible solution exists.

AN EXAMPLE IN DETAIL

The flows are, in what follows, quoted in L/s except where otherwise stated. The network shown in Fig. 3 was solved using the PHYAI method to illustrate the workings of the method in detail. It has ANIM

$$\boldsymbol{A} = \begin{pmatrix} 1 & -1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}$$

and

(a) all links have length 500 m, diameters 250 mm, roughnesses 0.03 mm,

(b) nodes 1 and 2 have zero elevation and the tank at node 3 has an elevation of 15 m,

(c) node 1 has demand 10 L/s and node 2 has demand 15 L/s,

(d) the flow in L/s through link 1 when no links are constrained is $q_1 = 2.0023$ L/s,

(e) the flow in link 1, the network's only co-tree link, is constrained to lie between zero and 1 L/s,

(f) the Wagner POR was chosen for this example (Wagner et al. 1988),

(g) the minimum pressure head, $h_m = 0$, and the service pressure head, $h_s = 20$ m.

The outflow constraints (6) for this case are $-c_{1,2} \leq 0$, $c_1 \leq 10$, $c_2 \leq 15$ and the linkflow constraints are $q_1 - 1 \leq 0$ and $-q_1 \leq 0$. Thus, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ each have two components and $\boldsymbol{w}_3$ and $\boldsymbol{w}_4$ each have one. At start all of the outflows are at the midpoints of their constraint intervals so none of their constraints are saturated: $c_1 = 5$, $c_2 = 7.5$. Thus, $\boldsymbol{S}_b^{(0)}$ in (18) is an identity matrix. The initial flow in link 1 is $q_1 = 0.5$ (the midpoint of

its constraint interval) so its constraint is also not saturated at start. The other two links have initial flows, q_j, $j = 1, 2$, given by $q_j = \frac{\pi}{12} D_j^2 = 16.4$ which correspond to velocities of 1/3 m/s. The initial point is therefore inside the problem's feasible region. The KKT conditions (12) to (17) now read

$$\begin{aligned}
\xi_1 - h_1 + h_2 - \frac{1}{t} v_1 &= 0 \\
\xi_2 + h_1 - 15 &= 0 \\
\xi_3 + h_2 - 15 &= 0 \\
h(c_1) - h_1 - \lambda_1 + \mu_1 &= 0 \\
h(c_2) - h_2 - \lambda_2 + \mu_2 &= 0 \\
-q_1 + q_2 - c_1 &= 0 \\
q_1 + q_3 - c_2 &= 0 \\
-\lambda_1 c_1 &= 0 \\
-\lambda_2 c_2 &= 0 \\
-\mu_1 (c_1 - d_1) &= 0 \\
-\mu_2 (c_2 - d_2) &= 0 \\
\lambda_{1,2} \geq 0, \quad \mu_{1,2} &\geq 0
\end{aligned}$$

The system (18) is, dropping iteration subscripts and noting that since no constraints are saturated the Lagrange multipliers $\lambda_{1,2} = \mu_{1,2} = 0$,

$$\left(\begin{array}{ccc|cc|cc}
f_1 & & & -1 & 1 & 0 & 0 \\
 & f_2 & & 1 & 0 & 0 & 0 \\
 & & f_3 & 0 & 1 & 0 & 0 \\
\hline
-1 & 1 & 0 & 0 & 0 & -1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & -1 \\
\hline
0 & 0 & 0 & -1 & 0 & h'(c_1) & \\
0 & 0 & 0 & 0 & -1 & & h'(c_2)
\end{array}\right)
\left(\begin{array}{c}
\delta q_1 \\ \delta q_2 \\ \delta q_3 \\ \hline \delta c_1 \\ \delta c_2 \\ \hline \delta h_1 \\ \delta h_2
\end{array}\right)
= -
\left(\begin{array}{c}
\xi_1 - h_1 + h_2 \\ \xi_2 + h_1 - 15 \\ \xi_3 + h_2 - 15 \\ \hline - q_1 + q_2 - c_1 \\ q_1 + q_3 - c_2 \\ \hline h(c_1) - h_1 \\ h(c_2) - h_2
\end{array}\right)$$

where the f_i represent the frictional head loss derivatives on the diagonal of the matrix $\boldsymbol{F}$ in (18). We remind the reader that $h(c)$ here is the inverse POR defined in (5).

The log barrier terms of (19) and (21) now need to be added to this system to get the system which is to be solved. Thus, f_1 is to be replaced by $\widehat{f}_1 = f_1 + \frac{1}{t} z_1 = f_1 + \frac{1}{t(q_1 - 1)^2} + \frac{1}{t q_1^2}$ and the first component of the right-hand-side vector now has $\frac{1}{t} v_1 = \frac{1}{t(q_1 - 1)} + \frac{1}{t q_1}$ subtracted

from it, giving the system to be solved at this step as

$$\left(\begin{array}{ccc|cc|cc}
f_1+\frac{1}{t}z_1 & & & \text{-1} & 1 & 0 & 0 \\
& f_2 & & 1 & 0 & 0 & 0 \\
& & f_3 & 0 & 1 & 0 & 0 \\
\hline
\text{-1} & 1 & 0 & 0 & 0 & \text{-1} & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & \text{-1} \\
\hline
0 & 0 & 0 & \text{-1} & 0 & h'(c_1) & \\
0 & 0 & 0 & 0 & \text{-1} & & h'(c_2)
\end{array}\right)
\left(\begin{array}{c}
\delta q_1 \\ \delta q_2 \\ \delta q_3 \\ \hline \delta c_1 \\ \delta c_2 \\ \hline \delta h_1 \\ \delta h_2
\end{array}\right)
= -
\left(\begin{array}{c}
\xi_1 - h_1 + h_2 - \frac{1}{t}v_1 \\ \xi_2 + h_1 - 15 \\ \xi_3 + h_2 - 15 \\ \hline - q_1 + q_2 - c_1 \\ q_1 + q_3 - c_2 \\ \hline h(c_1) - h_1 \\ h(c_2) - h_2
\end{array}\right)$$

The modification required to the system in (18) to get the system (25) for solution is thus seen to be minimal.

Rather than solve this system as a whole the smaller dimension system of (28) is solved for the heads, h_1, h_2, and then (29) and (30) are used to find the corresponding linkflows, q_1, and outflows, c_1, c_2. Once the updated unknowns are found using (31), the Lagrange multipliers, $\lambda_{1,2}$ and $\mu_{1,2}$ are found using (32). Of course, there are no Lagrange multipliers for the linkflow constraints.

The HYAI and ASMFC phases each took 4 iterations to satisfy the stopping test with a stopping tolerance of 10^{-14} when $t = 10^{10}$. The final heads were $h_1 = 14.92$ m and $h_2 = 14.88$ m. The final outflows were $c_1 = 8.64$ L/s and 12.94 L/s. The final linkflows $q_{1,2,3}$ were $1.00, 9.64, 11.94$ L/s, respectively so the flow constraint on link 1 was saturated at the upper bound. Its final Lagrange multiplier was $\nu_1 = 0.0378$. The set assignments did not change after the first iteration.

TABLES

Table 1: Convergence data for the HYAI method with continuation applied to the network shown in Fig. 4 .

k	e	$\delta_h^{(k)}$	$\delta_q^{(k)}$	$\delta_c^{(k)}$
1	12	$6.298900e-01$	$4.506700e-01$	$2.839100e-01$
2	12	$5.112000e-01$	$8.036900e-01$	$9.616300e-02$
3	12	$1.953100e-01$	$5.618000e+00$	$1.234900e-01$
4	12	$9.981000e-02$	$6.948300e-01$	$4.075300e-02$
5	12	$1.918700e-02$	$5.451400e-03$	$7.863300e-03$
6	12	$7.781600e-04$	$2.985700e-04$	$7.923300e-04$
1	13	$7.309900e-05$	$2.684000e-05$	$9.161300e-05$
1	14	$3.206800e-06$	$1.181400e-06$	$3.846000e-06$
1	15	$6.226300e-09$	$2.295200e-09$	$7.469500e-09$
1	16	$5.038500e-14$	$5.592700e-13$	$6.243100e-13$
1	17	$5.012200e-15$	$5.590300e-14$	$6.255000e-14$
1	18	$6.415700e-16$	$5.358600e-15$	$5.957200e-15$
1	19	$6.490100e-15$	$7.314800e-15$	$7.148600e-15$
1	20	$6.497500e-14$	$7.312400e-14$	$7.386900e-14$

Table 2: The parameters for the nine case study networks, the convergence results for the HYAI and PHYAI methods and the agreement between the solutions for the two methods.

						HYAI				PHYAI								
ID	n_p	n_j	n_f	n_c	$\zeta^{\%}$	k_h^h	σ_q^h	σ_h^h	σ_c^h	k_h^p	k_a^p	k_h^p $+k_a^p$	σ_q^p	σ_h^p	σ_c^p	α_q	α_h	α_c
N_1	934	848	8	84	69.8	12	13	16	18	11	3	14	13	16	18	14	8	12
N_2	1118	1039	2	74	15.3	15	12	16	19	12	3	15	11	12	18	14	10	12
N_3	1976	1770	4	205	51.8	23	11	16	17	15	5	20	11	16	17	14	9	10
$N_4^\dagger$	2465	1890	3	558	7.8	54	11	16	17	15	10	25	11	11	17	14	9	8
N_5	2508	2443	2	61	26.4	13	12	16	18	10	4	14	11	16	18	14	10	12
N_6	8584	8392	2	188	20.0	17	10	11	13	14	4	18	11	16	18	14	10	11
$N_7^\dagger$	14830	12523	7	1763	12.3	51	12	15	17	15	6	21	10	11	14	14	9	8
$N_8^\dagger$	19647	17971	15	1667	25.9	54	10	16	18	15	7	22	11	16	18	14	8	10
$N_9^\dagger$	157044	150630	4	5955	5.1	58	10	15	17	15	18	33	10	15	18	14	9	8

† required continuation to achieve a solution for $e = 20$.

Table 3: PHYAI and IPOPT iteration numbers and wall clock times compared.

ID	k_I	k_P	$C^*(\boldsymbol{q}, \boldsymbol{c})$	$\zeta\%$	s_I	s_P	s_I/s_P
N_1	33	14	−8.931	69.8	1.2	0.5	2.4
N_2	34	15	−0.229	15.3	0.7	0.3	2.3
N_3	51	20	−67.064	51.8	2.9	0.4	7.2
N_4	39	25	−169.188	7.8	0.9	0.8	1.1
N_5	16	14	−1.023	26.4	0.4	0.3	1.3
N_6	34	18	−1.690	20.0	4.3	1.2	3.6
N_7	50	21	−43.346	12.3	11.8	3.0	3.9
N_8	39	22	−27.563	25.9	17.5	4.9	3.6
N_9	80	33	−123.043	5.1	2016.0	221.7	9.1

Figure captions.

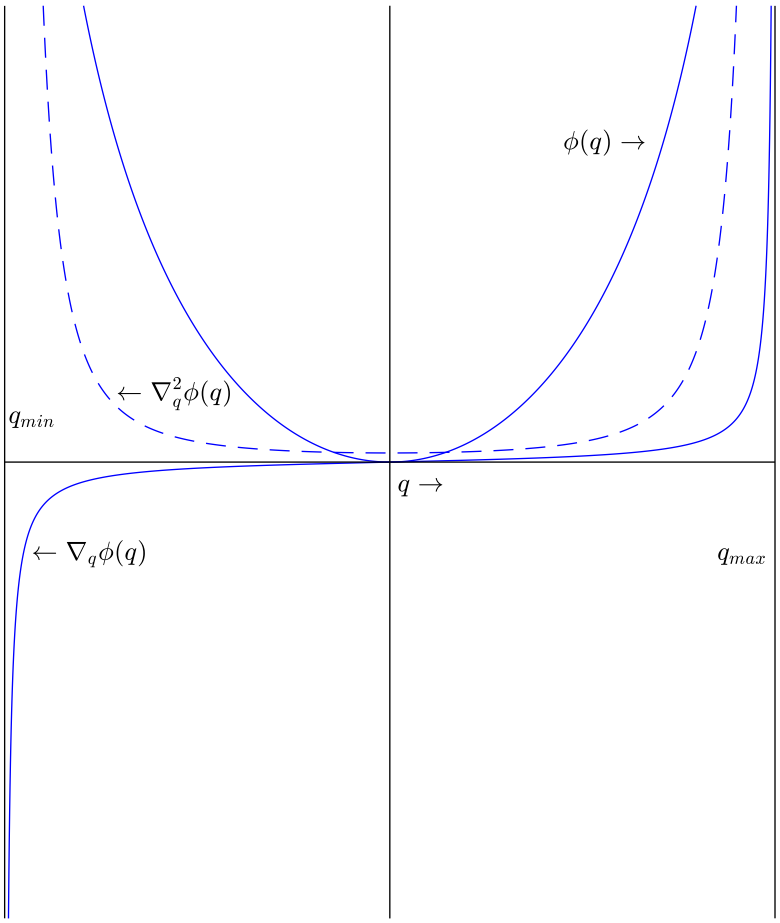

Figure 1: Representational plots of the log barrier function $\phi(q)$ of (7) and its first two derivatives.

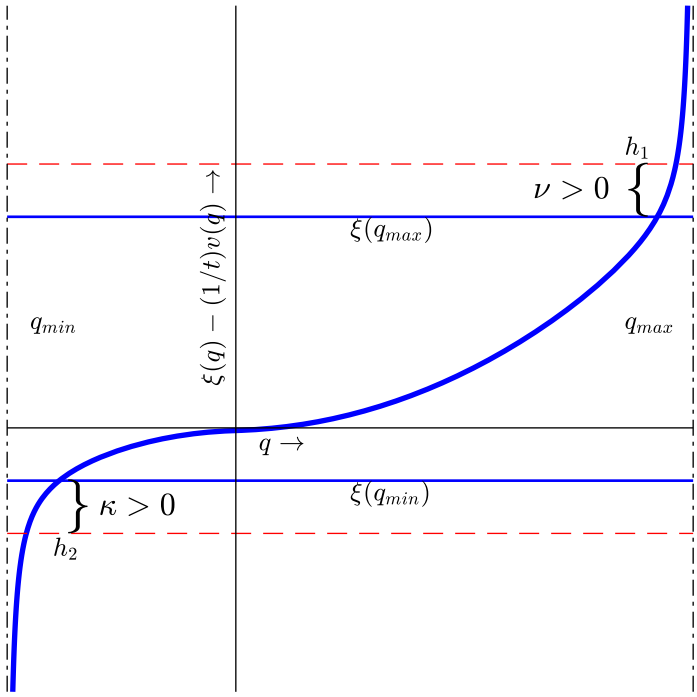

Figure 2: A representational plot of a function made up of a link head loss, $\xi(q)$, together with the corresponding log barrier term, $v(q)/t$.

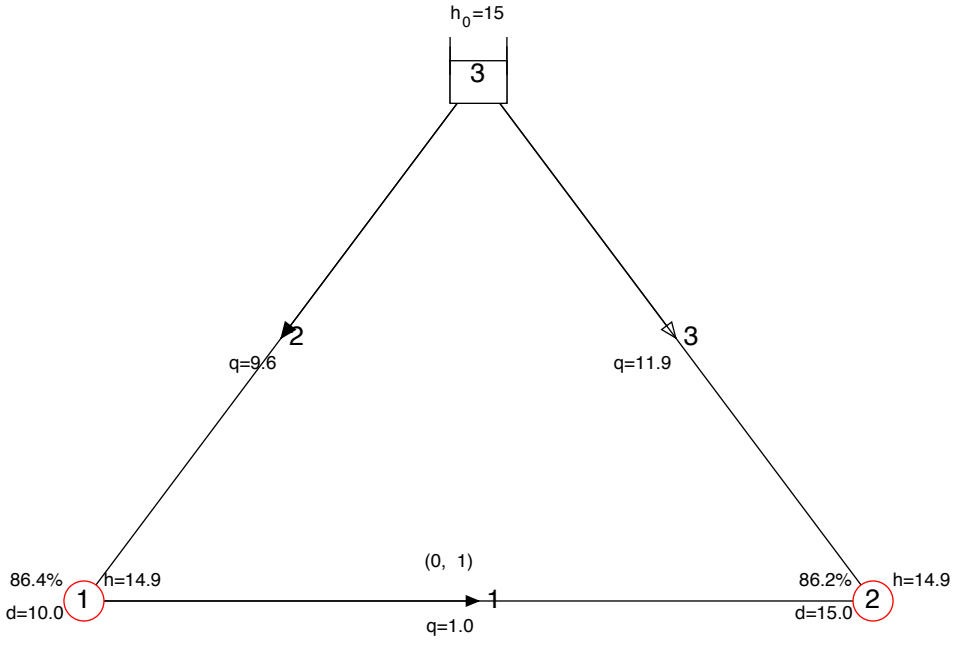

Figure 3: The figure for the network in the detailed example showing the solution heads, (m), linkflows (L/s) and delivery fractions. The linkflow constraint is shown in parentheses.

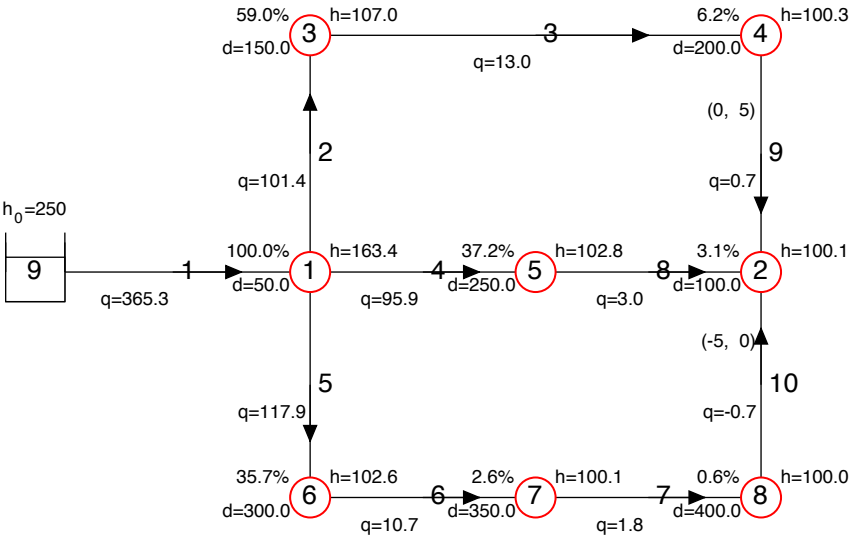

Figure 4: The (essentially) unconstrained solution for the small network discussed in the text.

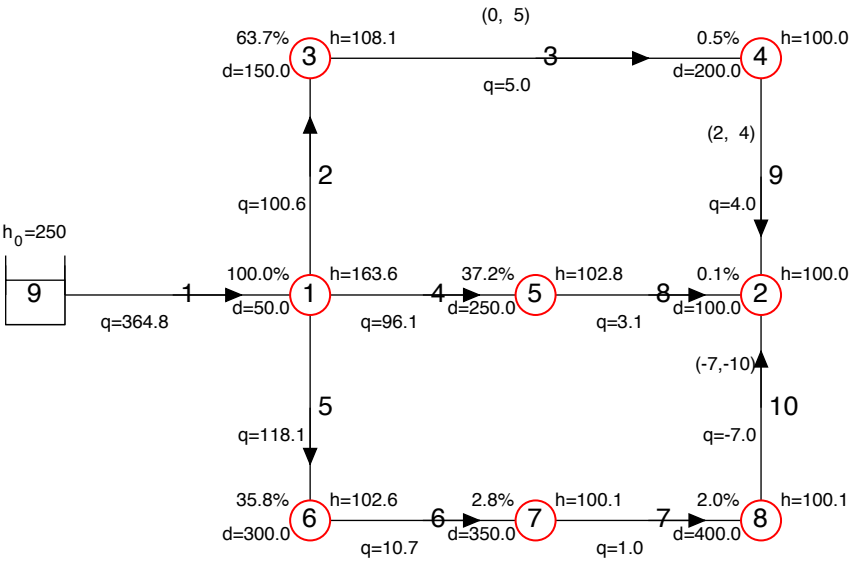

Figure 5: The constrained solution for the small network discussed in the text.